LETTER **Communicated by John Tsitsiklis**

Randomized Shortest-Path Problems: Two Related Models

Marco Saerens*
marco.saerens@uclouvain.be
Youssef Achbany
youssef.achbany@uclouvain.be
Information Systems Unit and Machine Learning Group, Université catholique de Louvain, Louvain-la-Neuve B-1348, Belgium

François Fouss
francois.fouss@uclouvain.be
Information Systems Unit and Machine Learning Group, Université catholique de Louvain, Louvain-la-Neuve B-1348, Belgium, and Management Sciences Department, Facultés Universitaires Catholiques de Mons, Mons 7000, Belgium

Luh Yen
luh.yen@uclouvain.be
Information Systems Unit and Machine Learning Group, Université catholique de Louvain, Louvain-la-Neuve B-1348, Belgium

This letter addresses the problem of designing the transition probabilities of a finite Markov chain (the policy) in order to minimize the expected cost for reaching a destination node from a source node while maintaining a fixed level of entropy spread throughout the network (the exploration). It is motivated by the following scenario. Suppose you have to route agents through a network in some optimal way, for instance, by minimizing the total travel cost—nothing particular up to now—you could use a standard shortest-path algorithm. Suppose, however, that you want to avoid pure deterministic routing policies in order, for instance, to allow some continual exploration of the network, avoid congestion, or avoid complete predictability of your routing strategy. In other words, you want to introduce some randomness or unpredictability in the routing policy (i.e., the routing policy is randomized). This problem, which will be called the *randomized shortest-path problem* (RSP), is investigated in this work. The global level of randomness of the routing policy is quantified by the expected Shannon entropy spread throughout the network and is provided a priori by the designer. Then, necessary conditions to compute the optimal randomized policy—minimizing

*Marco Saerens is also a research fellow of the IRIDIA Laboratory, Université Libre de Bruxelles.

the expected routing cost—are derived. Iterating these necessary conditions, reminiscent of Bellman's value iteration equations, allows computing an optimal policy, that is, a set of transition probabilities in each node. Interestingly and surprisingly enough, this first model, while formulated in a totally different framework, is equivalent to Akamatsu's model (1996), appearing in transportation science, for a special choice of the entropy constraint. We therefore revisit Akamatsu's model by recasting it into a sum-over-paths statistical physics formalism allowing easy derivation of all the quantities of interest in an elegant, unified way. For instance, it is shown that the unique optimal policy can be obtained by solving a simple linear system of equations. This second model is therefore more convincing because of its computational efficiency and soundness. Finally, simulation results obtained on simple, illustrative examples show that the models behave as expected.

1 Introduction

Algorithms for finding the shortest path in a network are currently used in a wide range of application areas, including transportation networks, medical imaging, wide-area network routing, and artificial intelligence, to name a few. Many extensions of the basic shortest-path algorithms have been proposed, still extending their application range (see, e.g., Bertsekas, 1998, 2000; Carre, 1979; Christofides, 1975; Jungnickel, 2004). In particular, many important applications developed in artificial intelligence, machine learning, pattern recognition, bioinformatics, and data mining, such as speech recognition (dynamic time warping or the Viterbi algorithm (Jelinek, 1997; Rabiner & Juang, 1993), sequence alignment (Durbin, Eddy, Krogh, & Mitchison, 1998; Gusfield, 1997), Markov decision processes (Bather, 2000; Puterman, 1994), routing in data networks (Bertsekas & Gallager, 1992), social network analysis (betweenness centrality; see Wasserman & Faust, 1994), game playing (minimax algorithms; see, e.g., Adelson-Velsky, Arlazarov, & Donskoy, 1988), planning (Ghallab, Nau, & Traverso, 2004; LaValle, 2006), neurocomputing (Bertsekas & Tsitsiklis, 1996), and reinforcement learning (Sutton & Barto, 1998; Powell, 2007) rely on one way or another on variants of shortest-path algorithms.

This work aims to introduce a related problem, which we call the randomized shortest-path problem (RSP), in the framework of a single source and a single destination. It can be described informally as follows. Suppose we have to find the path of minimum length from a source node to a destination node in a network, where the length of a path is the sum of the costs of the arcs on the path. Usually shortest-path algorithms provide pure deterministic routing policies: when standing in a given node k, we just follow the arc adjacent to k on the shortest path. In this letter, we investigate the possibility of randomizing the routing policy: the agents

could follow different paths according to some probability distribution. Of course, one may wonder why randomization should be introduced. For most problems, there is no advantage at all; however, there are some circumstances where randomization could eventually prove useful:

- If the environment is changing over time (nonstationary), the system could benefit from randomization or continual exploration. Indeed, without exploration, the agents are routed exclusively along the best path—without exploring alternative paths. They would therefore not be aware of the changes occurring in the network, for instance, some alternative path becoming shorter. For both models introduced in this letter, the structure of the network is supposed to be known, while the costs may change over time.
- Introducing randomness could be beneficial per se. Consider, for instance, the situation where an agent has to reach a given goal without being intercepted by some opponent. A deterministic shortest-path policy would make its behavior totally predictable; on the contrary, randomness introduces unpredictability and therefore renders interception more difficult. Randomization has proven useful for exactly this reason in game theory.
- When there are multiple destination nodes (or goals), introducing randomness allows the performance of some load balancing, by exploiting the goal nodes in parallel.
- Randomization also allows spreading the traffic on multiple paths, therefore reducing the danger of congestion. Indeed, when a randomized strategy is followed, the goods are routed along multiple paths and are therefore spread over the network.
- One may want to use a dissimilarity measure between two nodes that accounts for not only the shortest path, but also for all the other paths, with longer paths being penalized with respect to short ones, therefore considering that nodes connected by many short paths are closer than nodes connected by, for instance, only one short path (as in Fouss, Pirotte, Renders, & Saerens, 2007).
- In some application areas, such as sequence alignment, computing a similarity measure accounting for all paths could eventually provide better results than relying on the best path. This is a subject for further work.

For all these reasons, we decided to investigate randomized shortest-path problems. We thus define a randomized shortest-path problem as *a shortest-path problem to which an exploration constraint (e.g., an entropy constraint) is added in order to obtain optimal randomized policies* (also called stochastic policies), and therefore continual exploration.

Mathematically, randomization corresponds to the association of a probability distribution on the set of admissible arcs to follow in each node (choice randomization). If no randomization is present, the agents are

routed on the shortest path (the optimal choice) with probability one; only the best policy is exploited. Randomization appears when this probability distribution is no more peaked on the best choice: the agent is willing to sacrifice efficiency for exploration.

In this framework, we propose measuring the randomness associated to a given node by the (Shannon) entropy (see, e.g., Cover & Thomas, 2006; Kapur & Kesavan, 1992; Shannon, 1948) of the probability distribution on the set of admissible arcs to follow (transition probabilities) in this node. This entropy value captures the degree of randomness linked to the node. When the entropy is zero, there is no uncertainty, while when the entropy is maximal, a blind choice, with equal probability of following any arc, is performed.

Then, in a first model, we restate the randomized shortest-path problem as a global optimization problem: define the best randomized policy (the set of transition probabilities in each node) that minimizes the expected cumulated cost from the source node to the destination node while maintaining a fixed degree of randomness. This problem leads to a set of nonlinear equations defining necessary conditions of optimality. These equations, reminiscent of Bellman's equations, can be solved by iterating them until convergence. They provide a policy (the transition probabilities) that minimizes the expected cost from the initial node to the destination node, for a given degree of randomness. Interestingly enough, when the global degree of randomness is zero, the nonlinear equations reduce to Bellman's equations for finding the shortest path from the initial node to the destination node.

Then a second model, inspired by Akamatsu (1996) in transportation science (see section 1.1) will be studied and, while formulated in a totally different framework, will be shown to solve the same problem more efficiently for a special—but natural—choice of the entropy constraint. In particular, by recasting the problem into a statistical physics framework, it is shown that the randomized policy can be computed efficiently by solving a simple linear system of equations.

1.1 Related Work. The idea of quantifying the uncertainty linked to each node by entropy was introduced by Achbany, Fouss, Yen, Pirotte, and Saerens (2006, 2008) in the context of reinforcement learning and was inspired by the entropy rate of an ergodic Markov chain defined in information theory (see, e.g., Cover & Thomas, 2006). The main difference between this previous work is the fact that in this work, we fix the global entropy spread in the network instead of fixing the local entropy defined at each node in Achbany et al. (2006, 2008). While this difference seems a priori insignificant, it appears that constraining the global entropy spread into the network is more natural and more involved. Clearly, the nodes that need a large spread are difficult to determine in advance, and the model has to distribute the entropy by optimizing it globally all over the network. More

precisely, in this work, the global degree of randomness associated with the entire network is quantified by a weighted sum of the individual entropies associated with each node.

With the exception of this previous work (Achbany et al., 2006, 2008), to our knowledge, optimal randomized strategies, while popular, for instance in game theory (see, e.g., Osborne, 2004), and Markov games (Littman, 1994), have not been exploited in the context of shortest-path problems, with one noticeable exception, the work of Akamatsu (1996). Indeed, Akamatsu designed a randomized policy for routing traffic in transportation networks. In transportation science, randomized strategies are called *stochastic traffic assignments*, and within this context, Akamatsu's model is the model of reference. It provides a probability distribution of following an arc at any node of the network. More precisely, Akamatsu proposed to design the transition probabilities matrix in such a way that long paths to the destination are penalized. He therefore puts a Boltzmann distribution, depending on the length of the path, on the infinite set of paths, including paths containing cycles. He shows that this model results in a transition probabilities matrix that can be computed from the network by matrix inversion.

Surprisingly enough, his model solves randomized shortest-path problems, for a special choice of the entropy constraint, as shown in section 6. Indeed, Akamatsu did not notice that his algorithm solves a shortest-path problem. He simply proposed to weigh the paths in a heuristic way, without trying to optimize the total expected travel cost. In a second paper, Akamatsu (1997) proved that the total entropy spread in the network is a strictly concave function with respect to the arc flows and provided an interpretation of his model in terms of a different concept, the "expected minimum cost," also called "maximum utility," which plays an important role in random utility theory. This letter therefore provides a new interpretation for Akamatsu's model, as well as a new perspective, based on statistical physics, that allows us to derive the main results in a unified way.

Let us finally mention some interesting papers related to the present work. Nesterov (2007) introduces the notions of characteristic and potential functions of directed graphs and studies their properties. He applies his model to stochastic equilibrium traffic assignment problems. His framework is related to our work here, although the relationships are not evident. We plan to investigate the links between the two models in further work. On the other hand, the entropy of the paths (or trajectories) connecting an initial and an absorbing destination node of an absorbing Markov chain was studied by Ekroot and Cover (1993). These authors provided formulas for computing the entropy needed to reach the destination node. Tahbaz and Jadbabaie (2006) introduced a one-parameter family of algorithms that, as our algorithm, recovers both the Bellman-Ford procedure for finding shortest paths and the iterative algorithm for computing the average first-passage time. However, it was based on heuristic grounds, not on

a well-defined cost function to optimize. Moreover, it does not provide a randomized policy. In another context, Todorov (2006) studied a family of Markov decision problems that are linearly solvable, that is, for which a solution can be computed by solving a matrix eigenvector problem. In order to make this possible, Todorov assumes a special form for the control of the transition probabilities, which recasts the problem of finding the policy into an eigenvector problem. Boyd, Diaconis, and Xiao (2004) design a Markov chain that has the fastest mixing properties, and Sun, Boyd, Xiao, and Diaconis (2006) discuss its continuous-time counterpart. In a completely different framework, uninformed random walks, based on maximizing long-term entropy (Delvenne, 2005; Tomlin, 2003), have recently been proposed as an alternative of the standard PageRank algorithm. Finally, notice that some authors tackled the problem of designing ergodic (nonabsorbing) Markov or semi-Markov chains in a maximum entropy framework (see, e.g., Girardin, 2004; Girardin & Limnios, 2004, and the references there). This work is based on ergodic Markov chains, while our letter deals with absorbing Markov chains and randomized shortest-path problems.

1.2 Contributions and Organization of the Letter. In brief, this work has four contributions

- Randomized shortest-path problems are introduced and necessary conditions for solving them are derived (first model) for special choices of the entropy constraint. This may be considered a first step toward a more systematic treatment of such problems.
- The links between this first model and Akamatsu's stochastic traffic assignment model are studied; in particular, the two models are shown to be equivalent for a special choice of the entropy constraint. Thus, the randomized shortest-path model presented in this letter provides a new interpretation for Akamatsu's model: it is optimal in the sense that it minimizes the expected cost for a fixed entropy spread in the network.
- Akamatsu's model (1996) is revisited in a unified statistical physics formalism, which allows easily computing all the quantities of interest, in particular, the policy. It involves introducing a probability distribution on the infinite set of paths connecting the two nodes.
- It is shown that the optimal policy for Akamatsu's model can be obtained efficiently by solving a simple linear system of equations, which Akamatsu did not notice. This model therefore provides an alternative way of finding shortest paths in a network by solving a linear system of equations.

Since this work is somewhat theoretical, we also mention a few potential applications of randomized shortest paths (RSP) in artificial intelligence, machine learning, pattern recognition, bioinformatics, and data mining:

- *Routing and planning when the environment is changing*. Reinforcement learning and Markov decision algorithms based on the RSP could be designed and studied.
- *Using mixed, randomized, strategies based on the RSP instead of the usual minimax in game playing*. In a two-person game, it is unrealistic to assume that the opponent is completely rational, as minimax does. Therefore, it could be interesting to model the opponent's behavior by an RSP strategy instead of a pure minimax one, which leads to mixed minimax strategies.
- *Computing dissimilarities between nodes of a weighted, directed graph*. The expected cost for reaching one node from another node defines a dissimilarity measure between nodes of the graph, ranging from the shortest path to the average first-passage time distance, depending on the amount of entropy spread in the network. This idea was recently developed by Yen, Mantrach, Shimbo, and Saerens (2008) as an application of the models introduced in this letter. This fact can be exploited in data-mining applications such as recommender systems (see, e.g., Saerens, Fouss, Yen, & Dupont, 2004; Fouss et al., 2007). Indeed, random-walk or electrical-based proximity measures between nodes of a graph are becoming popular alternatives to the standard shortest-path distance (see Fouss et al., 2007; Klein & Randic, 1993; Nadler, Lafon, Coifman, & Kevrekidis, 2006; Qiu & Hancock, 2007; Saerens et al., 2004).
- *Computing dissimilarities between strings or sequences*. Instead of using the Viterbi algorithm on a lattice for computing the dissimilarity between two sequences, one could use the RSP algorithm, which will account for all the alignments between the two sequences instead of the single shortest one.

Of course, we do not know a priori if any of these applications of the RSP would be beneficial; this should essentially be considered propositions for further work. Moreover, we have to stress that the work presented here is not focused on any specific application area; rather, its main purpose is to study RSP problems and propose some general techniques to tackle them.

Section 2 introduces the notations, the randomized shortest-path problem, and the way we manage randomness. In section 3, the necessary conditions of optimality are derived. Section 4 describes an iterative procedure for computing the randomized policy. Section 5 restates Akamatsu's model in a statistical physics formalism and derives the main theoretical results obtained through this model. Section 7 shows that the first model introduced in section 3 is equivalent to Akamatsu's model in that it solves the same problem. Section 7 examines some computational issues, in particular, the case where the graph is acyclic. Section 8 shows some simulation examples, and section 9 presents the conclusion.

2 Statement of the Problem and Notations

For simplicity, this letter is focused on what is called *deterministic shortest-path problems*, as defined, for instance, by Bertsekas (2000).[1] Consider a weighted directed graph or network, G, with a set of n nodes V (or vertices) and a set of arcs E (or edges). To each arc linking node k and node k', we associate a number $c_{kk'}$ representing the immediate cost of following this arc. This cost can be positive (penalty), negative (reward), or zero, provided that no cycle exists whose total cost is negative (Christofides, 1975). In particular, this implies that if the graph is undirected, all costs are nonnegative. It is assumed that the whole network environment is known.

The choice to follow an arc from node k will be made according to a probability distribution (transition probabilities) defined on the set $S(k)$ of neighboring nodes (successors S) that can be reached from node k. These transition probabilities, defined on each node k, will be denoted as $p(k' \mid k) = p_{kk'}$ with $k' \in S(k)$. Furthermore, $\mathbf{P}$ will be the matrix containing the transition probabilities $p_{kk'}$ as elements. If there is no arc between k and k', we simply consider that $c_{kk'}$ takes a large value, denoted by ∞; in this case, the corresponding transition probability is set to zero, $p_{kk'} = 0$. The main difference between randomized and standard deterministic shortest-path problems resides in the fact that we will impose randomized choices. Randomization is introduced in order to guarantee a predefined degree of randomness that will be quantified by the Shannon entropy of the probability distributions. Randomized choices are common in a variety of fields, for instance, game theory (called mixed strategies in this context; see, e.g., Osborne, 2004), computer sciences (Motwani & Raghavan, 1995), Markov games (Littman, 1994), or decision sciences (Raiffa, 1970).

Moreover, like Bertsekas (2000), we assume there is a special cost-free destination or goal node d; once the system has reached that node, it remains there at no further cost. Thus, node d has no outgoing link except eventually d itself. In order to simplify the notations, we further assume in this work that we cannot return to the initial node k_0; that is, node k_0 has no incoming link (no predecessor). If this is not the case, just add a new initial node pointing to the previous one with zero cost. Finally, we will consider a problem structure such that termination is inevitable. Thus, the horizon is in effect finite, but its length is random, and it depends on the policy being used. The conditions for which this is true are, basically, related to the fact that the destination node can be reached in a finite number of steps from any potential initial node (for a rigorous treatment, see Bertsekas, 2000, or Bertsekas & Tsitsiklis, 1996).

[1] Notice that in this work, paths may contain cycles. Paths containing cycles are also commonly called *walks* in the literature.

The overall goal will be to determine the set of probability distributions $\pi \equiv \{p(k' \mid k); k, k' = 1, 2, \ldots, n\}$, contained in the transition probabilities matrix $\mathbf{P}$, and called the choice probabilities or the policy, that minimizes

$$v_\pi(k_0) = \mathbb{E}_\pi \left\{ \sum_{t=0}^{\infty} c_{s_t s_{t+1}} \mid s_0 = k_0 \right\}, \tag{2.1}$$

where s_t is a random variable containing the label of the node reached at time step t and $v_\pi(k_0)$ is the total expected cost accumulated over an infinite horizon, when starting from the initial (or source) node k_0, and following policy π. The expectation is taken with respect to the transition probabilities associated with the nodes.

Thus, the main objective is to design a randomized policy minimizing the expected cost-to-go, equation 2.1, subject to an entropy constraint controlling the total randomness spread in the network, and therefore the exploration rate. In other words, we are looking for an optimal policy, $\pi^* = \text{argmin}_\pi[v_\pi(k_0)]$ or, equivalently, an optimal transition probabilities matrix $\mathbf{P}^*$ subject to an entropy constraint. Stated differently, the problem is to design an optimal finite, first-order Markov chain minimizing the expected cost needed to reach a destination state from an initial state, while fixing the entropy spread in the chain.

2.1 Total Expected Cost and Markov Chains. Remember that the essence of the problem is to reach the destination node $s = d$ with minimal expected cost. Once the set of transition probabilities, π, is fixed, this problem can be represented as a first-order finite Markov chain where each node is a state. The destination state is then considered as absorbing with no outgoing link. In this framework, the problem of computing the expected cost (see equation 2.1) from any state k is closely related to the computation of the "average first-passage time" in the associated Markov chain (Kemeny & Snell, 1976; Norris, 1997). The average first-passage time is the average number of steps a random walker starting from the initial state k_0 will take in order to reach destination state d for the first time. By first-step analysis (see, e.g., Taylor & Karlin, 1998), one can easily show that once the transition probabilities are fixed, the total expected cost $v_\pi(k)$ can be computed through the following equations:

$$\begin{cases} v_\pi(k) = \displaystyle\sum_{k' \in S(k)} p_{kk'}[c_{kk'} + v_\pi(k')], & \text{for } k \neq d \\ v_\pi(d) = 0, & \text{for destination state } d \end{cases}, \tag{2.2}$$

where the sum is taken on the successor nodes of k, $S(k)$. These equations can be iterated in order to find the expected costs; a closed-form solution will be derived later and necessitates introducing of the fundamental matrix of the Markov chain.

Let us now renumber the states in order to have state k_0 in the first position (index 1) and state d in the last position (index n). After this reordering, the formula can be put in matrix form:

$$\mathbf{v} = \mathbf{diag}(\mathbf{PC}^{\mathbf{T}}) + \mathbf{Pv}, \tag{2.3}$$

where $\mathbf{diag}(\mathbf{M})$ is a column vector containing the elements on the diagonal of matrix $\mathbf{M}$, $\mathbf{v}$ is a n-dimensional column vector containing the expected costs-to-go, $v_\pi(k)$, for each of the $(n-1)$ first states, and a 0 as the last (nth) element, while $\mathbf{C}$ is the matrix containing the immediate costs $c_{kk'}$ (with $c_{nn} = 0$). Of course, the superscript T denotes the matrix transpose.

From this Markov chain, one can also compute the expected number of visits to each state in the following way (Kemeny & Snell, 1976; Norris, 1997). First, observe that the states of the Markov chain can be divided into two families: the transient states, $k \in \{1, \ldots, n-1\}$ and one absorbing state, n. The transition probabilities matrix $\mathbf{P}$ can therefore be rewritten in partitioned form,

$$\mathbf{P} = \begin{bmatrix} \mathbf{Q} & \mathbf{r} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}, \tag{2.4}$$

where $\mathbf{Q}$ is the $(n-1) \times (n-1)$ substochastic matrix of transition probabilities among the transient states, $\mathbf{r}$ is an $(n-1) \times 1$ column vector representing the transition probabilities from transient states to the absorbing state n, and $\mathbf{0}$ is an $(n-1) \times 1$ column vector full of 0's. It is well known that the probability distribution of finding a random walker, starting from state 1 at $t = 0$, in any state at time step t, is provided by vector $\mathbf{x}(t) = (\mathbf{P}^{\mathrm{T}})^t \mathbf{e}_1$, where $\mathbf{x}(t)$ is an $n \times 1$ column vector with components $x_i(t)$ corresponding to the probability that the random walker is in state $s = i$ at time step t and $\mathbf{e}_1$ is a $n \times 1$ column vector containing a 1 as the first element and 0's otherwise. When the analysis is focused on the transient states only, the corresponding probability of finding the random walker in any transient state at time step t is $\widetilde{\mathbf{x}}(t) = (\mathbf{Q}^{\mathrm{T}})^t \widetilde{\mathbf{e}}_1$, where $\widetilde{\mathbf{x}}(t)$ and $\widetilde{\mathbf{e}}_1$ are $(n-1) \times 1$ column vectors obtained from $\mathbf{x}(t)$ and $\mathbf{e}_1$ by removing the last (nth) element.

Therefore, the expected number of visits to each transient state, when starting from state 1 at time $t = 0$, is provided by

$$\mathbf{n} = \sum_{t=0}^{\infty} \widetilde{\mathbf{x}}(t) = \sum_{t=0}^{\infty} (\mathbf{Q}^{\mathrm{T}})^t \widetilde{\mathbf{e}}_1 = (\mathbf{I} - \mathbf{Q}^{\mathrm{T}})^{-1} \widetilde{\mathbf{e}}_1 = \mathbf{N}^{\mathrm{T}} \widetilde{\mathbf{e}}_1. \tag{2.5}$$

The matrix $\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1}$ is usually called the fundamental matrix of the Markov chain (Kemeny & Snell, 1976; Norris, 1997). Its elements $n_{ij} = [\mathbf{N}]_{ij}$ correspond to the expected number of times the process is in transient state

j if it is initiated in state i. Thus, the column vector $\mathbf{n}$ contains the expected number of visits to each transient state, when starting from state 1.

In section A.1 in the appendix, we show that the expected costs $\mathbf{v}$ can be expressed in closed form in terms of the fundamental matrix $\mathbf{N}$. Indeed, by partitioning the cost matrix $\mathbf{C}$, containing the immediate costs $c_{kk'}$, as

$$\mathbf{C} = \begin{bmatrix} \mathbf{D} & \mathbf{s} \\ \boldsymbol{\infty}^{\mathrm{T}} & 0 \end{bmatrix}, \tag{2.6}$$

equation 2.3 can be solved in terms of $\mathbf{v}$ and reexpressed as (see section A.1)

$$\widetilde{\mathbf{v}} = \mathbf{N}\,\mathbf{diag}(\mathbf{Q}\mathbf{D}^{\mathrm{T}} + \mathbf{r}\mathbf{s}^{\mathrm{T}}), \tag{2.7}$$

where $\widetilde{\mathbf{v}}$ is a $(n-1)\times 1$ column vector containing the $(n-1)$ first elements of $\mathbf{v}$ (the nth element being trivially equal to 0). These relationships will be useful later when deriving the necessary conditions of optimality.

2.2 Controlling Randomness by Fixing the Entropy Spread in the Network. Now that we have introduced the problem, we will explain how we manage the exploration. In each state $k \neq n$, we compute the Shannon entropy (Cover & Thomas, 2006; Kapur & Kesavan, 1992; Shannon, 1948) of the transition probabilities:

$$h_k = -\sum_{k' \in S(k)} p_{kk'} \log p_{kk'}, \quad \text{with} \quad k \neq n. \tag{2.8}$$

In matrix form, equation 2.8 becomes

$$\mathbf{h} = -\mathbf{diag}(\mathbf{Q}(\log \mathbf{Q})^{\mathrm{T}}), \tag{2.9}$$

where the logarithm is taken elementwise. $\mathbf{h}$ is an $(n-1)\times 1$ column vector containing the h_k for each transient state (for the last, absorbing, state n, the entropy is trivially equal to 0). Thus, h_k measures the uncertainty about the choice in state k. It is equal to zero when there is no uncertainty at all ($p_{kk'}$ reduces to a Kronecker delta); it is equal to $\log(m_k)$, where m_k is the number of admissible choices (outdegree) at node k, in the case of maximum uncertainty ($p_{kk'} = 1/m_k$; a uniform distribution). Increasing the entropy increases randomness; a maximum entropy aims to a completely random choice since the next state is chosen completely at random, with a uniform distribution, without taking the costs into account.

The global (weighted) entropy H, measuring the degree of randomness spread in the whole network, is simply defined as the sum of the individual

entropy values in each state h_k, weighted by some factor u_k:

$$H = \sum_{k=1}^{n-1} u_k h_k = -\sum_{k=1}^{n-1} u_k \sum_{k' \in S(k)} p_{kk'} \log p_{kk'}. \tag{2.10}$$

The factors u_k weigh the contribution of each node to the global entropy and could depend on the transition probabilities. This quantity is similar to the entropy rate defined for stationary (or ergodic) Markov chains, in which case the weights u_k are equal to the stationary distribution of the Markov chain (see, e.g., Shannon, 1948; Cover & Thomas, 2006). The main difference here is that we are dealing with absorbing Markov chains, which are therefore not ergodic.

The next step is to set the global entropy H to a predefined value, say, $H = H_0$, therefore fixing the global randomness occurring in the network, and to compute the optimal transition probabilities, minimizing the total expected cost, $v_\pi(1)$.

3 Optimal Routing Policy Under Entropy Constraint

We now turn to the problem of determining an optimal policy under a global entropy constraint. More precisely, we seek the set of transition probabilities, $\pi \equiv \{p_{kk'}\}$, for which the expected cost v_π (see equation 2.1) from initial state 1 to destination state n is minimal while fixing the global entropy in the network, H, to a constant value. It can be formulated as a constrained optimization problem involving a Lagrange function: minimize v_π (see equation 2.1) subject to the constraint $H = H_0$ (H being given by equation 2.10). In other words, we seek $\mathbf{P}^* = \arg\min_{\mathbf{P}}[v_\pi]$ (see equation 2.1) subject to the constraint $\sum_{k=1}^{n-1} u_k h_k = H_0$.

In section A.2 in the appendix, we derive the form of the optimal transition probabilities distribution within state k, which appears to be a multinomial logit, or Boltzmann, distribution:

$$p_{kk'} = \frac{\exp\left[-\frac{n_k}{\eta\, u_k}(c_{kk'} + v(k')) + \frac{1}{u_k}\sum_{l \neq n}(\partial u_l / \partial p_{kk'}) h_l\right]}{\sum_{l' \in S(k)} \exp\left[-\frac{n_k}{\eta\, u_k}(c_{kl'} + v(l')) + \frac{1}{u_k}\sum_{l \neq n}(\partial u_l / \partial p_{kl'}) h_l\right]}, \quad \text{for } k \neq n, \tag{3.1}$$

where n_k is the kth element of $\mathbf{n}$ (containing as entries the first row of the fundamental matrix $\mathbf{N}$—see equation 2.5) and $v(k)$ is the minimum expected cost computed thanks to equation 2.7. The parameter $\eta > 0$ controls the entropy: the larger η, the larger the entropy. Notice that η could be found from the relationship $H_0 = -\sum_{k=1}^{n-1} u_k \sum_{k' \in S(k)} p_{kk'} \log p_{kk'}$ by using, for instance, a bisection algorithm (the relation between η and H is

monotonically decreasing, all other parameters being fixed), but this is not needed since η can be provided by the user in place of H_0. In the sequel, it will therefore be assumed that the user provides the value of the parameter η instead of H_0.

3.1 Fixing the Expected Entropy Spread in the Network. By looking to equation 3.1, we observe that when a constant weight $u_k = 1$ is assumed, the entropy is preferably spread on the states that are seldom visited (having a small n_k). This behavior is intuitively not very appealing since the global entropy does not properly reflect the amount of randomness in this case. Therefore, a convenient choice would be to weigh the entropy related to each node, h_k, by the expected number of visits to this node, that is, to set $u_k = n_k$,

$$H = \sum_{k=1}^{n-1} n_k h_k = -\sum_{k=1}^{n-1} n_k \sum_{k' \in S(k)} p_{kk'} \log p_{kk'}. \tag{3.2}$$

Interestingly enough, this particular choice will be shown to be similar to Akamatsu's model, which will be introduced in section 5. The global entropy H can be interpreted as the expected total entropy spread in the network for a random walker traveling from node 1 to node n. We finally have to compute the second term $\sum_{l \neq n} (\partial u_l / \partial p_{kk'}) h_l = \sum_{l \neq n} (\partial n_l / \partial p_{kk'}) h_l$ in equation 3.1, which is done in section A.3 in the appendix:

$$\sum_{l \neq n} (\partial n_l / \partial p_{kk'}) h_l = n_k \widetilde{\mathbf{e}}_{k'}^{\mathrm{T}} \mathbf{N} \mathbf{h} = n_k \sum_{l \neq n} n_{k'l} h_l, \tag{3.3}$$

where $n_{k'l}$ is element (k', l) of matrix $\mathbf{N}$ and $\widetilde{\mathbf{e}}_k$ is an $(n-1) \times 1$ column vector full of 0's except its kth entry containing a 1. Let us rewrite equation 3.1 by setting $u_k = n_k$, and using equation 3.3,

$$\begin{aligned} p_{kk'} &= \frac{\exp\left[-\frac{1}{\eta} (c_{kk'} + v(k')) + \varkappa_{k'} \right]}{\sum_{l' \in S(k)} \exp\left[-\frac{1}{\eta} (c_{kl'} + v(l')) + \varkappa_{l'} \right]}, \quad \text{with } k \neq n \\ \text{where } &\begin{cases} \varkappa_{k'} = \sum_{l \neq n} n_{k'l} h_l, & \text{for } k' \neq n \\ \varkappa_n = 0, & \text{for destination state } n \end{cases}. \end{aligned} \tag{3.4}$$

Notice that the second term in the exponential of the numerator of equation 3.4, $\varkappa_{k'}$, can be interpreted as the expected entropy when starting from state k'. On the other hand, the first term in the exponential, $(c_{kk'} + v(k'))$, corresponds to the expected cost when deciding to jump to node k' from node k. This term is weighted by $1/\eta$ and is counterbalancing the expected entropy. η plays the role of a temperature. When the global entropy H is

small, that is, η is small as well, the first term in the exponential dominates the second one, which can be neglected in this special case. Moreover, when $\eta \to 0$, the largest $p_{kk'}$ will dominate the other terms in the exponential of equation 3.4 and $\varkappa_{k'} \to 0$, with the result that the computation of the expected cost, equation 2.2, reduces to Bellman's equations for finding the shortest path from the initial state to the destination state in this case.

Equations 2.7 and 3.4 are thus the necessary optimality conditions. The parameter η is supposed to be provided a priori by the user, according to the desired degree of randomness he or she is willing to concede: in this work, no attempt has been made to optimize or estimate it. This would, however, be an interesting research topic for further work.

3.2 Fixing the Expected Entropy per Visit. Yet another sensible choice would be to fix the expected entropy per visit,

$$H = \frac{\sum_{k=1}^{n-1} n_k h_k}{\sum_{l=1}^{n-1} n_l} = -\sum_{k=1}^{n-1} \pi_k \sum_{k' \in S(k)} p_{kk'} \log p_{kk'}, \tag{3.5}$$

where $\pi_k = n_k / \sum_{l=1}^{n} n_l$. Some calculus, similar to the previous computation, shows that the resulting necessary conditions are

$$\begin{aligned} p_{kk'} &= \frac{\exp\left[-\frac{1}{\eta}(c_{kk'} + v(k')) + \varkappa_{k'} \right]}{\sum_{l' \in S(k)} \exp\left[-\frac{1}{\eta}(c_{kl'} + v(l')) + \varkappa_{l'} \right]}, \quad \text{with} \quad k \neq n \\ \text{where} \quad & \begin{cases} \varkappa_{k'} = \sum_{l \neq n} n_{k'l} h_l - \left(\sum_{l \neq n} n_{k'l} \right) \left(\sum_{l' \neq n} \pi_{l'} h_{l'} \right), & \text{for } k' \neq n \\ \varkappa_n = 0, & \text{for destination state } n \end{cases} \end{aligned} \tag{3.6}$$

Other choices are, of course, possible, depending on the problem at hand.

3.3 Using a Reference A Priori Policy by Fixing the Küllback Divergence. In this section, we show that we could start from a reference, a priori, policy, denoted by $p_{kk'}^{\text{ref}}$, and fix the Küllback divergence J (see, e.g., Kapur & Kesavan, 1992) between this reference policy and the policy we are seeking. Thus, we define j_k as

$$j_k = \sum_{k' \in S(k)} p_{kk'} \log \frac{p_{kk'}}{p_{kk'}^{\text{ref}}} \tag{3.7}$$

and J as

$$J = \sum_{k=1}^{n-1} u_k j_k = \sum_{k=1}^{n-1} u_k \sum_{k' \in S(k)} p_{kk'} \log \frac{p_{kk'}}{p^{\text{ref}}_{kk'}}. \tag{3.8}$$

By proceeding as in the previous section, we easily obtain

$$p_{kk'} = \frac{p^{\text{ref}}_{kk'} \exp\left[-\frac{n_k}{\eta\, u_k}(c_{kk'} + v(k')) + \frac{1}{u_k}\sum_{l \neq n}(\partial u_l / \partial p_{kk'})\, j_l\right]}{\sum_{l' \in S(k)} p^{\text{ref}}_{kl'} \exp\left[-\frac{n_k}{\eta\, u_k}(c_{kl'} + v(l')) + \frac{1}{u_k}\sum_{l \neq n}(\partial u_l / \partial p_{kl'})\, j_l\right]}, \quad \text{for } k \neq n. \tag{3.9}$$

This time, a sensible choice could be to weigh the entropy related to each node, h_k, by the expected number of visits to this node when using the reference policy, n^{ref}_k; that is, to set $u_k = n^{\text{ref}}_k$. This leads to the following necessary conditions:

$$p_{kk'} = \frac{p^{\text{ref}}_{kk'} \exp\left[-\frac{n_k}{\eta\, n^{\text{ref}}_k}(c_{kk'} + v(k'))\right]}{\sum_{l' \in S(k)} p^{\text{ref}}_{kl'} \exp\left[-\frac{n_k}{\eta\, n^{\text{ref}}_k}(c_{kl'} + v(l'))\right]}, \quad \text{for } k \neq n. \tag{3.10}$$

4 Computation of the Optimal Policy

Equations 2.7 and 3.4 suggest a simple iterative procedure, similar to the well-known value-iteration algorithm, for the computation of the policy. This first, iterative, algorithm computes the optimal policy while maintaining the expected entropy spread in the network. It will be used in all our experiments:

Algorithm 1
Computation of the optimal policy while maintaining the expected entropy spread in the network: a first, iterative, algorithm.
Input:

- Node 1 is the initial node while node n is the destination node. The absorbing node n can be reached from any other node of the network.
- $\eta > 0$: the parameter controling the degree of exploration.
- $\mathbf{C} = \begin{bmatrix} \mathbf{D} & \mathbf{s} \\ \boldsymbol{\infty}^{\mathrm{T}} & 0 \end{bmatrix}$: the $n \times n$ cost matrix; node n is the destination node.

1. Initialize $\mathbf{P} \leftarrow \begin{bmatrix} \mathbf{Q} & \mathbf{r} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}$, for instance, by setting $p_{kk'} = \frac{1}{m_k}$ $(k \neq n)$ where m_k is the outdegree of node k.
2. **repeat**
3. **if** matrix $(\mathbf{I} - \mathbf{Q})$ is not of full rank **then**
4. **return** Error: the fundamental matrix is not invertible.
5. **end if**

6. $\mathbf{h} \leftarrow -\mathbf{diag}(\mathbf{Q}(\log\mathbf{Q})^{\mathrm{T}})$ {see equation 2.9}.
7. Solve $(\mathbf{I}-\mathbf{Q})\widetilde{\boldsymbol{\varkappa}} = \mathbf{h}$ with respect to $\widetilde{\boldsymbol{\varkappa}}$ {see equation 3.4}.
8. $\boldsymbol{\varkappa} \leftarrow \begin{bmatrix} \widetilde{\boldsymbol{\varkappa}} \\ 0 \end{bmatrix}$
9. Solve $(\mathbf{I}-\mathbf{Q})\widetilde{\mathbf{v}} = \mathbf{diag}(\mathbf{Q}\mathbf{D}^{\mathrm{T}} + \mathbf{r}\mathbf{s}^{\mathrm{T}})$ with respect to $\widetilde{\mathbf{v}}$ {see equation 2.7}.
10. $\mathbf{v} \leftarrow \begin{bmatrix} \widetilde{\mathbf{v}} \\ 0 \end{bmatrix}$
11. $$\begin{cases} p_{kk'} \leftarrow \dfrac{\exp\left[-\frac{1}{\eta}\left(c_{kk'} + v(k')\right) + \varkappa_{k'}\right]}{\sum_{l' \in S(k)} \exp\left[-\frac{1}{\eta}\left(c_{kl'} + v(l')\right) + \varkappa_{l'}\right]}, & \text{for all } k \neq n, k' \\ p_{nk'} = \delta_{nk'}, \text{ for destination node } n \end{cases}$$ {see equation 3.4}.
12. **until** convergence of $\mathbf{P}$.
13. **return** the policy $\mathbf{P}$ of the form $\begin{bmatrix} \mathbf{Q} & \mathbf{r} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}$: the transition probabilities matrix containing the elements $p_{kk'}$.

Notice that instead of computing the fundamental matrix, we prefer to solve two linear systems of equations at each iteration.

The algorithm is obtained by performing a block coordinate descent on the following Lagrange function:

$$\begin{aligned} \mathcal{L} = v(1) &+ \sum_{k \neq n} \lambda_k \left[v(k) - \sum_{k' \in S(k)} p_{kk'} \left(c_{kk'} + v(k')\right) \right] \\ &+ \lambda_n [v(n) - 0] + \sum_{k \neq n} \mu_k \left[\sum_{k' \in S(k)} p_{kk'} - 1 \right] \\ &+ \eta \left[\sum_{k \neq n} n_k \sum_{k' \in S(k)} p_{kk'} \log p_{kk'} + H_0 \right] \\ &+ \sum_{k' \neq n} \phi_{k'} \left[n_{k'} - \sum_{k \neq n} p_{kk'} n_k - \delta_{1k'} \right], \end{aligned} \tag{4.1}$$

which is identical to the Lagrange function defined in section A.2, equation A.4, except that the formula for computing the expected number of passages, n_k, is made explicit, so that the n_k can now be considered as independent variables. Indeed, $\mathbf{n} = \mathbf{N}^{\mathrm{T}}\mathbf{e}_1$, from which we deduce $\mathbf{n} = \mathbf{Q}^{\mathrm{T}}\mathbf{n} + \mathbf{e}_1$ and thus $n_{k'} = \sum_{k \neq n} p_{kk'} n_k - \delta_{1k'}$. Notice also that the η parameter related to the temperature is fixed a priori (it is not considered as a variable) and is provided by the user. The necessary conditions of optimality are the same

as for the previous formulation detailed in section A.2 and displayed in equation 3.4.

Setting the partial derivative of $\pounds$ in terms of μ_k and $p_{kk'}$ equal to zero while fixing all the other parameters provides the update equations for the transition probabilities matrix $\mathbf{P}$, which is uniquely attained. On the other hand, setting the partial derivative in terms of the other parameters (except η, considered as fixed) equal to zero provides the reestimation equations for the expected costs $\mathbf{v}$ and the expected entropies $\boldsymbol{\varkappa}$ (we easily find $\phi_k = \eta\varkappa_k$), which are both uniquely attained provided the matrix $(\mathbf{I} - \mathbf{Q})$ is invertible. These two steps are iterated until convergence (see algorithm 1). It is known that a block coordinate descent algorithm converges to a stationary point (a local minimum) for continuously differentiable objective functions provided the individual minimum for each block of variables is uniquely attained (see, e.g., Bertsekas, 1999), which is indeed the case. Moreover, we show in section 6 that the second model (Akamatsu's model) solves the same optimization problem whose solution is shown to be unique in this new formalism. Based on this fact, we provide in section 6.3 an informal proof showing that the stationary point obtained by algorithm 1 must be a global minimum.

Computationally, it is very demanding since it requires, in addition to being iterative, either the computation of the fundamental matrix, $\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1}$, or the solution of two linear systems of equations at each iteration. Section 5 introduces a much more efficient procedure to determine the optimal policy, which is therefore recommended instead of algorithm 1. However, in changing environments, one could initialize the policy thanks to the procedure introduced in section 5 and then rely on local iterative updating rules such as algorithm 1.

On the other hand, the decision for quantifying randomness by the expected entropy spread through the network is also questionable; alternative solutions are also conceivable, such as other measures of entropy (see Kapur & Kesavan, 1992) or simply taking the sum of the variances of the transition probabilities over all the nodes.

5 Randomized Shortest Paths Based on Akamatsu's Model

In this section, we introduce Akamatsu's model (Akamatsu, 1996; referred to as the second model) and restate it into a sum-over-paths statistical-physics formalism (see, e.g., Jaynes, 1957; Reichl, 1998; Schrödinger, 1952), allowing us to compute the quantities of interest in a unified way. This approach can therefore be considered as a kind of discrete counterpart of the well-known continuous-time path integral introduced by Feynman (1948).

5.1 A Statistical-Physics Framework for Akamatsu's Model. The idea behind this second model is to define a Markov chain for which each path $\wp_r$ from the initial state to the destination state has a probability of being

followed proportional to $\exp[-\theta E(\wp_r)]$, where $E(\wp_r) = E_r$ is the total cost associated with the rth path, and referred to as the energy associated to that path. In other words, short paths are favored, and long paths are penalized. The parameter θ is supposed to be provided a priori by the user, according to the desired degree of randomness he is willing to concede.

As for the first model (see equation 2.1), we consider that the total cost associated with a path is additive, that is,

$$E(\wp_r) = \sum_{t=0}^{t_f} c_{s_t s_{t+1}}, \tag{5.1}$$

where $s_0 = k_0$ is the initial state and $s_{t_f} = d$ is the destination state. Here, we assume that $\wp_r$ is a valid path from the initial state to the destination state, that is, all $c_{s_t s_{t+1}} \neq \infty$ along that path and all $s_t \neq d$, except the last state, s_{t_f}, which is equal to d, $s_{t_f} = d$. Remember also that the destination state d is made absorbing so that an infinite cost is associated with each transition from this state. In other words, once the random walker has reached this state, he disappears with no additional cost. The probability of following the path $\wp_r$ is thus

$$\mathrm{P}(\wp_r) = \frac{\exp[-\theta E(\wp_r)]}{\sum_{r=1}^{\infty} \exp[-\theta E(\wp_r)]}. \tag{5.2}$$

The set of all paths connecting the starting state k_0 and the destination state d (appearing only once on the path—node d is absorbing), on which the probability distribution, equation 5.2, is defined, will be denoted by $\mathcal{R}$.

Let us now show that all the quantities of interest can be computed from a quantity, $\mathcal{Z} = \sum_{r=1}^{\infty} \exp[-\theta E(\wp_r)]$, the denominator of equation 5.2, which corresponds to the partition function in statistical physics (see, e.g., Jaynes, 1957; Reichl, 1998; Schrödinger, 1952) or Markov random fields (Rue & Held, 2005). The main point is that the partition function can easily be computed from the immediate cost matrix $\mathbf{C}$ (see the next section). We further define the free energy F as $F = -\frac{1}{\theta}\log(\mathcal{Z}) = -T\log(\mathcal{Z})$, where $T = 1/\theta$ is the temperature of the system.

Let us first compute the expected energy or cost needed to reach the destination state from the initial state in terms of the partition function:

$$\overline{E} = \frac{\partial(-\log(\mathcal{Z}))}{\partial\theta} = \frac{\sum_{r=1}^{\infty} \exp\left[-\theta E(\wp_r)\right] E(\wp_r)}{\sum_{r'=1}^{\infty} \exp\left[-\theta E(\wp_{r'})\right]} = \sum_{r=1}^{\infty} \mathrm{P}(\wp_r) E(\wp_r). \tag{5.3}$$

Moreover, it can easily be shown (Jaynes, 1957) that the moment $m > 1$ of the energy can be found by

$$\mathbb{E}\{(E - \overline{E})^m\} = (-1)^m \frac{\partial^m}{\partial \theta^m}(\log(\mathcal{Z})), \tag{5.4}$$

where $\mathbb{E}\{.\}$ is the expectation operator.

Now, the expected number of passages through the link $k \to k'$ can also be easily computed:

$$\begin{aligned}\overline{\eta}_{kk'} &= \frac{\partial F}{\partial c_{kk'}} = \frac{1}{\theta}\frac{\partial(-\log(\mathcal{Z}))}{\partial c_{kk'}} = \sum_{r=1}^{\infty}\frac{\exp\left[-\theta E(\wp_r)\right]}{\mathcal{Z}}\delta(r;k,k') \\ &= \sum_{r=1}^{\infty}\mathrm{P}(\wp_r)\delta(r;k,k'),\end{aligned} \tag{5.5}$$

where $\delta(r;k,k')$ denotes the number of times the link $k \to k'$ is present in path number r, $\wp_r$, and thus the number of times the link is traversed. The conditional probability of following the link $k \to k'$, that is, the transition probability $p_{kk'}$ if the resulting Markov process is a first-order Markov chain, is then provided by

$$p_{kk'} = \frac{\sum_{r=1}^{\infty}\mathrm{P}(\wp_r)\delta(r;k,k')}{\sum_{l=1}^{n}\sum_{r=1}^{\infty}\mathrm{P}(\wp_r)\delta(r;k,l)} = \frac{\overline{\eta}_{kk'}}{\sum_{l=1}^{n}\overline{\eta}_{kl}}. \tag{5.6}$$

On the other hand, the expected number of passages in state k is

$$\overline{n}_k = \sum_{l=1}^{n}\overline{\eta}_{lk}, \tag{5.7}$$

which corresponds to the expected number of incoming visits. Notice that the expected number of visits to state k was simply denoted by n_k (without a overline) in section 3.

The total entropy spread in the network can be computed as well. First, define the entropy H of the paths $\wp_r \in \mathcal{R}$ as (Ekroot & Cover, 1993)

$$\begin{aligned}H &= -\sum_{r=1}^{\infty}\mathrm{P}(\wp_r)\log\mathrm{P}(\wp_r) \\ &= -\sum_{r=1}^{\infty}\frac{\exp\left[-E(\wp_r)/T\right]}{\mathcal{Z}}\log\left(\frac{\exp\left[-E(\wp_r)/T\right]}{\mathcal{Z}}\right)\end{aligned}$$

$$= -\sum_{r=1}^{\infty} \frac{\exp\left[-E(\wp_r)/T\right]}{\mathcal{Z}} \left(-\frac{E(\wp_r)}{T} - \log \mathcal{Z}\right)$$

$$= \log \mathcal{Z} + \frac{1}{T} \sum_{r=1}^{\infty} E(\wp_r) \frac{\exp\left[-E(\wp_r)/T\right]}{\mathcal{Z}} = \log \mathcal{Z} + \frac{\overline{E}}{T}. \tag{5.8}$$

The entropy H can also be found by

$$H = -\frac{\partial F}{\partial T} = \log \mathcal{Z} + \frac{T}{\mathcal{Z}} \frac{\partial \mathcal{Z}}{\partial T}$$

$$= \log \mathcal{Z} + \sum_{r=1}^{\infty} \frac{E(\wp_r)}{T} \frac{\exp\left[-E(\wp_r)/T\right]}{\mathcal{Z}} = \log \mathcal{Z} + \frac{\overline{E}}{T}, \tag{5.9}$$

which is indeed the same as equation 5.8.

5.2 Computation of the Partition Function $\mathcal{Z}$. By following Akamatsu's argument (Akamatsu, 1996), let us now show how $\mathcal{Z}$ can be computed from the cost matrix. We start from the immediate cost matrix, $\mathbf{C}$, from which we build a new matrix, $\mathbf{W}$, which simply contains the exponentials of the costs c_{ij}, that is, of the elements of $\mathbf{C}$,

$$\mathbf{W} = \exp[-\theta \mathbf{C}], \tag{5.10}$$

where the exponential is taken elementwise. Now, we easily observe that element (k_0, d) of the matrix $\mathbf{W}^t$ ($\mathbf{W}$ to the power t) is $[\mathbf{W}^t]_{k_0 d} = \sum_{r \in \mathcal{R}(t)} \exp\left[-\theta E(\wp_r)\right]$ where $\mathcal{R}(t)$ is the set of paths connecting the initial node k_0 to the destination node d in exactly t transition steps. Consequently, the partition function is

$$\mathcal{Z} = \sum_{t=1}^{\infty} \sum_{r \in \mathcal{R}(t)} \exp\left[-\theta E(\wp_r)\right] = \left[\sum_{t=1}^{\infty} \mathbf{W}^t\right]_{k_0 d}, \tag{5.11}$$

which converges if the spectral radius of $\mathbf{W}$, $\rho(\mathbf{W})$, is less than 1 and depends on the parameter θ. We will therefore assume in the sequel that $\rho(\mathbf{W}) < 1$. Since the matrix $\mathbf{W}$ contains only nonnegative elements, a sufficient condition for $\rho(\mathbf{W}) < 1$ is that all its row sums are less than 1, which can always be achieved for a sufficiently large value of θ (see equation 5.10) and thus a sufficiently low entropy. Indeed, it is well known that the spectral radius of a real square matrix is always smaller than or equal to its maximum absolute row sum norm (see, e.g., Bronson, 1989).

More generally, element k of the dth column of the matrix $\sum_{t=1}^{\infty}\mathbf{W}^t$ corresponds to the partition function when starting from an initial state k. We are therefore interested in the dth column of this matrix.

Computing this infinite series for the problem at hand is relatively easy. By reordering the states in such a way that the initial state corresponds to the first state (state 1) and the destination state to the last state (state n), the matrix $\mathbf{W}$ takes the following form:

$$\mathbf{W} = \begin{bmatrix} \mathbf{Q}_w & \mathbf{r}_w \\ \mathbf{0}^{\mathrm{T}} & 0 \end{bmatrix}. \tag{5.12}$$

Indeed, the elements of the last row are set to $\exp[-\theta \ \infty] = 0$ ($c_{nk'} = \infty$); we do not allow any transition from the last (nth) absorbing state and therefore put a 0 on every entry of this last row. Computing the series of powers of $\mathbf{W}$ provides

$$\sum_{t=1}^{\infty}\mathbf{W}^t = (\mathbf{I}-\mathbf{W})^{-1} - \mathbf{I}. \tag{5.13}$$

Thus, assuming that the initial and the destination states are different ($n \neq 1$), $\mathcal{Z}$ can be computed thanks to

$$\begin{aligned} \mathcal{Z} &= [(\mathbf{I}-\mathbf{W})^{-1} - \mathbf{I}]_{1n} = \mathbf{e}_1^{\mathrm{T}}[(\mathbf{I}-\mathbf{W})^{-1} - \mathbf{I}]\mathbf{e}_n \\ &= \mathbf{e}_1^{\mathrm{T}}(\mathbf{I}-\mathbf{W})^{-1}\mathbf{e}_n = [\mathbf{Z}]_{1n} = z_{1n}, \end{aligned} \tag{5.14}$$

where we posed $\mathbf{Z} = (\mathbf{I}-\mathbf{W})^{-1} = \mathbf{I} + \mathbf{W} + \mathbf{W}^2 + \cdots$. Notice that since the last row of $\mathbf{W}$ is $\mathbf{0}^{\mathrm{T}}$, so are the powers of $\mathbf{W}$, so that $z_{nn} = 1$. Matrix $\mathbf{Z}$ will be called the fundamental matrix by analogy with the theory of absorbing Markov chains (see the definition of matrix $\mathbf{N}$ after equation 2.5). Notice that if the cost matrix $\mathbf{C}$ is symmetric (undirected network), since it is assumed that $\rho(\mathbf{W}) < 1$, the matrix $(\mathbf{I}-\mathbf{W})$ is positive definite.

5.3 Computation of the Main Quantities. We now have to compute the derivatives of $\mathcal{Z}$ (assuming a paths' probability distribution defined by equation 5.2) in terms of θ and $c_{kk'}$ in order to obtain the different quantities of interest (see equations 5.3 to 5.8), which is done in section A.4 in the appendix. For the expected energy or cost, we obtain

$$\overline{E} = \frac{\partial(-\log \mathcal{Z})}{\partial\theta} = -\frac{\mathbf{e}_1^{\mathrm{T}}\mathbf{Z}\mathbf{W}'_{\theta}\mathbf{Z}\mathbf{e}_n}{z_{1n}}, \tag{5.15}$$

where the matrix $\mathbf{W}'_\theta$ contains the elements $[\mathbf{W}'_\theta]_{ij} = -c_{ij}\exp[-\theta c_{ij}]$. The expected number of transitions through the link $k \to k'$ is

$$\overline{\eta}_{kk'} = \frac{\partial F}{\partial c_{kk'}} = \frac{z_{1k} z_{k'n} \exp\left[-\theta c_{kk'}\right]}{z_{1n}}, \quad \text{with} \quad k \neq n. \tag{5.16}$$

The expected number of passages through state k is

$$\overline{n}_k = \frac{z_{1k} z_{kn}}{z_{1n}}, \quad \text{with} \quad k \neq n. \tag{5.17}$$

Notice that the expected number of passages through state k was simply denoted by n_k (without a overline) in section 3. The transition probabilities (see equation 5.6) are provided by

$$p_{kk'} = \frac{z_{k'n}}{z_{kn}} \exp\left[-\theta c_{kk'}\right], \quad \text{with} \quad k \neq n. \tag{5.18}$$

And finally, the entropy spread in the network is

$$H = -\frac{\partial F}{\partial T} = \log \mathcal{Z} + \frac{\overline{E}}{T} = \log z_{1n} + \theta \overline{E}. \tag{5.19}$$

6 Links Between Akamatsu's Model and the First Randomized Shortest-Path Model

This section shows that Akamatsu's model presented in section 5 and the randomized shortest-path model constrained by the expected entropy spread in the network (see section 3.1) are in fact equivalent. By equivalent, we mean that both formulations solve the same constrained optimization problem, namely, respectively, (1) the two objective functions (see section 6.1) and (2) the two constraints on the entropy spread in the network (see section 6.2) refer to the same quantities. Thus, both models minimize the same objective function, subject to the same entropy equality constraint, with respect to either the transition probabilities (first model; see section 3) or the paths probabilities (second model; see section 6.3). The first model thus computes an optimal set of transition probabilities, while the second one computes an optimal set of paths probabilities. Indeed, in section 6.3, it is shown that the optimal policy, that is, the optimal paths probability distribution, is a Boltzmann distribution, as assumed in Akamatsu's model (see equation 5.2).

6.1 Equivalence of the Objective Function. As objective function, the first model minimizes the expected cost expressed as $v_\pi(1) = \mathbb{E}_\pi\{\sum_{t=0}^{\infty} c_{s_t s_{t+1}} | s_0 = k_0\}$ (see equation 2.1), the value of this expected cost being provided by the recurrence relation of equation 2.2. We show in this section that the cost functions and the expectation operators are equivalent in both formulations (first and second model).

Since the cost of remaining in the absorbing destination state is zero and the energy of a trajectory in the second model (Akamatsu) is defined as $E(\wp_r) = \sum_{t=0}^{t_f} c_{s_t s_{t+1}}$ (see equation 5.1), the expected cost v_π (see equation 1.1) can be reexpressed in terms of the paths probability distribution:

$$\begin{aligned} v_\pi(1) &= \mathbb{E}_\pi \left\{ \sum_{t=0}^{\infty} c_{s_t s_{t+1}} \mid s_0 = k_0 \right\} = \mathbb{E}_\pi \left\{ \sum_{t=0}^{t_f} c_{s_t s_{t+1}} \mid s_0 = k_0 \right\} \\ &= \mathbb{E}_\pi \left\{ E(\wp_r) \right\}, \ \wp_r \text{ starting in } k_0 \text{ and ending in } d \\ &= \sum_{r=1}^{\infty} \Pr(\wp_r) E(\wp_r), \ \wp_r \in \mathcal{R}, \end{aligned} \tag{6.1}$$

where, from the (first-order) Markov property,

$$\Pr(\wp_r) = \prod_{k,k'} (p_{kk'})^{\delta(r;k,k')}, \tag{6.2}$$

is the probability of a trajectory $\wp_r \in \mathcal{R}$ in the formalism of the first model. Remember that $\delta(r;k,k')$ corresponds to the number of times the link $k \to k'$ is present in path number r, $\wp_r$. We thus end up in equation 6.1 with the expected energy, as defined in equation 5.3, provided $\Pr(\wp_r) = \mathrm{P}(\wp_r)$ (where $\mathrm{P}(\wp_r)$ is given by equation 5.2), which is shown in the next paragraph. Notice that it is assumed here that the model is a first-order Markov chain. It will be shown in section 6.3 that the second (Akamatsu's) model minimizes the expected cost of equation 6.1.

Let us now prove that the probability of a path is equivalent in both formulations, that is, $\Pr(\wp_r) = \mathrm{P}(\wp_r)$ when equation 5.18 is true and considering a first-order Markov chain. A similar proof was already provided by Akamatsu (1996) in his formulation of the problem. We have

$$\begin{aligned} \Pr(\wp_r) &= \prod_{k,k'} (p_{kk'})^{\delta(r;k,k')} \\ &= \prod_{k,k'} \left[\frac{z_{k'n}}{z_{kn}} \exp\left[-\theta c_{kk'}\right] \right]^{\delta(r;k,k')} \\ &= \frac{\prod_{k,k'} \left[z_{k'n}\right]^{\delta(r;k,k')}}{\prod_{k,k'} \left[z_{kn}\right]^{\delta(r;k,k')}} \prod_{k,k'} \left[\exp\left[-\theta c_{kk'}\right]\right]^{\delta(r;k,k')} \\ &= \frac{z_{nn}}{z_{1n}} \exp\left[-\theta \sum_{k,k'} c_{kk'} \delta(r;k,k') \right] \\ &= \frac{1}{\mathcal{Z}} \exp\left[-\theta E(\wp_r)\right] = \mathrm{P}(\wp_r), \end{aligned} \tag{6.3}$$

where we used equation 5.18 as well as $E(\wp_r) = \sum_{k,k'} c_{kk'}\delta(r;k,k')$, $z_{nn} = 1$ and $z_{1n} = \mathcal{Z}$. This shows that the paths probability distribution, equation 5.2, corresponds to (can be generated by) a first-order Markov chain. This is, however, not true in general for arbitrary paths distributions since higher-order Markov chains also lead to paths probability distributions on $\mathcal{R}$, but not necessarily of the form 5.2. Thus, the objective function—the expected cost-to-go—is equivalent in both models.

6.2 Equivalence of the Entropy Concept. We now show that the two definitions of entropy, namely equations 3.2 and 5.8, express the same quantity. In a second paper, Akamatsu (1997) proved that the entropy defined by equation 5.8 can be decomposed into the sum of two terms: a link-based and a node-based term. Here, we adapt his proof, as well as the work on the entropy rate of a Markov chain (Cover & Thomas, 2006), in order to show the equivalence between the entropy concepts. We easily find

$$\begin{aligned}
H &= -\sum_{r=1}^{\infty} \mathrm{P}(\wp_r)\log \mathrm{P}(\wp_r) \\
&= -\sum_{r=1}^{\infty} \mathrm{P}(\wp_r)\log\left[\prod_{k,k'}(p_{kk'})^{\delta(r;k,k')}\right] \\
&= -\sum_{r=1}^{\infty} \mathrm{P}(\wp_r)\sum_{k,k'=1}^{n}\delta(r;k,k')\log p_{kk'} \\
&= -\sum_{k,k'=1}^{n}\left[\sum_{r=1}^{\infty}\mathrm{P}(\wp_r)\delta(r;k,k')\right]\log p_{kk'} \\
&= -\sum_{k,k'=1}^{n}[\overline{\eta}_{kk'}]\log p_{kk'} = -\sum_{k,k'=1}^{n}\overline{n}_k\, p_{kk'}\log p_{kk'} \\
&= -\sum_{k=1}^{n-1}\overline{n}_k\sum_{k'=1}^{n} p_{kk'}\log p_{kk'}. \qquad (6.4)
\end{aligned}$$

Notice that we used $\overline{\eta}_{kk'} = \overline{n}_k\, p_{kk'}$ as well as the definitions introduced in section 5.1. We also needed $0\log 0 = 0$ and $0^0 = 1$. Now, equation 6.4 is exactly the entropy defined in equation 3.2. Thus, the two notions of entropy are equivalent.

6.3 Equivalence of the Optimization Problems. We now restate the problem of finding the optimal policy in the second, sum-over-paths, framework. The policy in this second model corresponds to the set of paths probabilities instead of the set of transition probabilities in the first model.

The objective is then to find the set of paths probabilities, $\{\mathrm{P}(\wp_r)\}$ (the paths probability distribution), minimizing $\overline{E} = \sum_{r=1}^{\infty} \mathrm{P}(\wp_r) E(\wp_r)$ subject to the constraint $-\sum_{r=1}^{\infty} \mathrm{P}(\wp_r) \log \mathrm{P}(\wp_r) = H_0$. By defining the Lagrange function,

$$\mathcal{L} = \sum_{r=1}^{\infty} \mathrm{P}(\wp_r) E(\wp_r) + \lambda \left[\sum_{r=1}^{\infty} \mathrm{P}(\wp_r) \log \mathrm{P}(\wp_r) + H_0 \right] + \mu \left[\sum_{r=1}^{\infty} \mathrm{P}(\wp_r) - 1 \right], \tag{6.5}$$

we, not surprisingly, find the Boltzmann distribution,

$$\mathrm{P}(\wp_r) = \frac{\exp\left[-\theta E(\wp_r)\right]}{\sum_{r=1}^{\infty} \exp\left[-\theta E(\wp_r)\right]} = \frac{1}{\mathcal{Z}} \exp\left[-\theta E(\wp_r)\right], \tag{6.6}$$

with $\theta = 1/\lambda$, as expected. This Boltzmann distribution corresponds to a unique, global minimum, and it is clear that there is no other stationary point. Actually equation 6.6 corresponds exactly to the paths probability distribution that was assumed in Akamatsu's model (see equation 5.2). Notice that if the constraint involves Küllback-Leibler's divergence with respect to a reference policy, as in section 3.3, instead of the Shannon entropy, the whole framework remains valid (see Yen et al., 2008, for a derivation of the main quantities in this context).

Let us now recapitulate the arguments. For the first model, the optimal transition probabilities $p_{kk'}$ are directly found by minimizing the objective function provided by equation 2.1, subject to an entropy constraint (see equation 2.10). For the second (sum-over-paths) model, the optimal paths probabilities $\mathrm{P}(\wp_r)$ are found instead by minimizing the objective function of equation 5.3, subject to the entropy constraint $H_0 = -\sum_{r=1}^{\infty} \mathrm{P}(\wp_r) \log \mathrm{P}(\wp_r)$. The solution for this second formulation is unique and corresponds to the Boltzmann distribution (see equation 6.6); moreover, there is no other stationary point. Therefore, this paths probability distribution induces a directed flow (average number of passages) passing through each arc, given by equation 5.16. This arc flow is also unique provided that $\rho(\mathbf{W}) < 1$. In the case of a near-zero entropy, if there exist several shortest paths between the initial and the destination node, according to equation 6.6, the traffic is distributed equally (with uniform probability) over these shortest paths.

The transition probabilities are simply obtained by dividing the arc flow by the total outgoing flow out of the node (see equations 5.6, 5.18). Consequently, the second, sum-over-paths, model induces unique transition probabilities and thus a unique first-order Markov model. The parameter

$1/\eta$ in the first model (see equation 3.4) plays exactly the same role as the parameter θ in the sum-over-paths model (see equation 5.2).

Stated differently, the first model computes optimal transition probabilities while the second (Akamatsu's) model computes optimal paths probabilities. Since, for a first-order Markov chain, paths probabilities can be deduced from transition probabilities (see equation 6.2) and vice versa (see equation 5.6), both formulations are equivalent. This shows that Akamatsu's model presented in section 5 and the randomized shortest-path model developed in section 3.1 solve the same problem by two alternative ways.

Furthermore, let us now return to the first model and show heuristically that the set of transition probabilities, $\pi^* = \{p^*_{kk'}\}$, obtained by algorithm 1 and being a stationary point of $v_\pi(1)$, corresponds to a global minimum, at least when $\rho(\mathbf{W}) < 1$. First, denote as $\mathcal{P}_1$ the set of paths probability distributions that are equivalent to first-order Markov chains (by equivalent, we mean that they can be computed from a first-order Markov chain by equation 6.2). The set $\mathcal{P}_1$ is a restriction of the complete, unrestricted set containing all the paths probability distributions, denoted by $\mathcal{P}$ (including also those generated by higher-order Markov chains). Now, it is clear from the above discussion (see equation 6.3) that the global minimum of $\overline{E}$ in terms of the $\mathrm{P}(\wp_r)$ belonging to the complete set $\mathcal{P}$, and provided by equation 6.6, also belongs to $\mathcal{P}_1$. Since the objective function admits only one stationary point (the global minimum) in $\mathcal{P}$, and $\mathcal{P}_1$ is a restriction of $\mathcal{P}$, this global optimum is also the unique stationary point within $\mathcal{P}_1$ (see the discussion following equation 6.6). Furthermore, we already know that π^* is a stationary point of $v_\pi(1)$ (see section 4). The corresponding paths probabilities, $\{\mathrm{P}^*(\wp_r)\}$, obtained from π^* by equation 6.2, must also be a stationary point of $\overline{E}$ among the restricted set $\mathcal{P}_1$ as the two objective functions are equivalent. Since there is no stationary value other than the global minimum, $\{\mathrm{P}^*(\wp_r)\}$ must be a global minimum, and so does π^*. This reasoning is, however, not a rigorous, formal proof since most of the arguments are rather heuristic and the support of the paths probability distributions is infinite in the second model. A rigorous treatment of the properties of the solution provided by algorithm 1 is outside the scope of this work.

Note finally that the problem can also be restated as a maximum entropy problem as introduced by Jaynes (1957): maximize the entropy spread in the network while maintaining a given expected cost.

7 Some Computational Issues When Computing Akamatsu's Model for Large Cyclic and Acyclic Networks

7.1 Computational Issues. When dealing with large graphs, the inversion of $(\mathbf{I} - \mathbf{W})$ can be a serious issue. However, by examining equations 5.15 to 5.19, we immedialy notice that only $\mathbf{Z}^{\mathrm{T}}\mathbf{e}_1 = \mathbf{z}_1$ and $\mathbf{Z}\mathbf{e}_n = \mathbf{z}_n$

need to be computed. These two quantities can be found by solving the linear systems of equations,

$$(\mathbf{I} - \mathbf{W})^{\mathrm{T}}\mathbf{z}_1 = \mathbf{e}_1 \quad \text{and} \quad (\mathbf{I} - \mathbf{W})\mathbf{z}_n = \mathbf{e}_n. \tag{7.1}$$

Notice that these equations could also be solved iteratively by the updating equations $\mathbf{z}_1 \leftarrow \mathbf{W}^{\mathrm{T}}\mathbf{z}_1 + \mathbf{e}_1$ and $\mathbf{z}_n \leftarrow \mathbf{W}\mathbf{z}_n + \mathbf{e}_n$ since we assume $\rho(\mathbf{W}) < 1$.

In other words, the column vector $\mathbf{z}_1 = (\mathbf{row}_1(\mathbf{Z}))^{\mathrm{T}}$ contains the elements of the first row of matrix $\mathbf{Z}$, while the column vector $\mathbf{z}_n = \mathbf{col}_n(\mathbf{Z})$ contains the elements of the last column of $\mathbf{Z}$. These linear systems of equations can be solved efficiently, especially when the matrix $\mathbf{W}$ is sparse (Davis, 2006), which is often the case. Remember also that for undirected graphs, the matrix $(\mathbf{I} - \mathbf{W})$ is positive definite.

Elementwise, these last equations yield

$$\begin{cases} z_{11} = 1 \\ z_{1k'} = \displaystyle\sum_{k \in P(k')} \exp[-\theta c_{kk'}] z_{1k} \end{cases} \tag{7.2}$$

and

$$\begin{cases} z_{nn} = 1 \\ z_{kn} = \displaystyle\sum_{k' \in S(k)} \exp[-\theta c_{kk'}] z_{k'n}, \end{cases} \tag{7.3}$$

where, as before, $P(k')$ is the set of predecessors of node k' and $S(k)$ is the set of successors of node k. Equations 7.2 and 7.3 actually provide an intuitive interpretation of the forward and backward variables z_{1k}, z_{kn} when the maximum row sum of both matrices $\mathbf{W}$ and $\mathbf{W}^{\mathrm{T}}$ is less than 1. Consider a special random walk defined by the transition probabilities matrix $\mathbf{W}$ with absorbing state n. Since $\mathbf{W}$ is substochastic, the random walker has a nonzero probability of disappearing at each time step. In this case, the values z_{kn} in equation 7.3 can be interpreted as the probability of reaching node n for a random walker starting in node k (see, e.g., Kemeny & Snell, 1976). In a symmetric way, the transition probabilities matrix $\mathbf{W}^{\mathrm{T}}$ also defines a random walk, with absorbing state 1 this time, and a similar interpretation for the z_{1k} values can be developed.

Moreover, there is an interesting similarity between equations 5.16 and 7.2–7.3 and the forward and backward procedure for computing the forward and backward variables when estimating transition probabilities of a hidden Markov model (Jelinek, 1997; Rabiner & Juang, 1993). These equations are also similar to the estimation equation obtained for conditional random fields (Lafferty, McCallum, & Pereira, 2001). Our results are, however, more general since in our case, the graph contains cycles. The

algorithm is also quite similar in spirit to the ant colony optimization algorithm (Dorigo & Stutzle, 2004), where the amount of pheronome dropped at each node k is represented by z_{kn}.

Consequently, we obtain the algorithm displayed in algorithm 2, allowing computation of the transition probabilities matrix $\mathbf{P}$. Remember that we assume that the user supplies the parameter θ in place of the entropy value H_0:

Algorithm 2
Computation of the optimal policy while maintaining the expected entropy spread in the network: a second, more efficient, algorithm inspired by Akamatsu's model.
Input:

- Node 1 is the initial node while node n is the destination node. The absorbing node n can be reached from any other node of the network.
- $\theta > 0$: the parameter controlling the degree of exploration.
- $\mathbf{C} = \begin{bmatrix} \mathbf{D} & \mathbf{s} \\ \boldsymbol{\infty}^{\mathrm{T}} & \infty \end{bmatrix}$: the $n \times n$ cost matrix.

1. $\mathbf{W} = \exp[-\theta \mathbf{C}]$ {Elementwise exponential; see equation 5.10}
2. **if** $\rho(\mathbf{W}) \geq 1$ **then**
3. **return** Error: the spectral radius is greater than one.
4. **end if**
5. Solve $(\mathbf{I} - \mathbf{W})\mathbf{z}_n = \mathbf{e}_n$ with respect to $\mathbf{z}_n$. The elements of $\mathbf{z}_n$ are z_{kn}. {see equation 7.1}
6. Compute $\begin{cases} p_{kk'} = \dfrac{z_{k'n}}{z_{kn}} \exp[-\theta c_{kk'}], & \text{for all } k \neq n, k' \\ p_{nk'} = \delta_{nk'}, & \text{for destination node } n \end{cases}$ {see equation 5.18}
7. **return** the policy $\mathbf{P}$ of the form $\begin{bmatrix} \mathbf{Q} & \mathbf{r} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}$: the transition probabilities matrix containing the elements $p_{kk'}$.

7.2 Dealing with Acyclic Networks. We now show that the computation of the optimal policy is greatly simplified when dealing with acyclic networks or lattices. An acyclic network is a network for which there is no cycle, that is, one can never return to the same node. An interesting procedure allowing simplification of a network with one source node and one destination node in order to obtain an acyclic network has been proposed by Dial (1971).

It is clear that if the graph is directed and acyclic, the matrix $(\mathbf{I} - \mathbf{W})$ is upper triangular, and the matrix $(\mathbf{I} - \mathbf{W})^{\mathrm{T}}$ is lower triangular after reordering the nodes according to a topological ordering (the topological ordering or sorting of a directed acyclic graph is a linear ordering of its nodes in which each node comes before all nodes to which it has outgoing edges; see, e.g., Sedgewick, 1990). In this special case, the linear equations defined in equations 7.1 can easily be solved by simple backsubstitution.

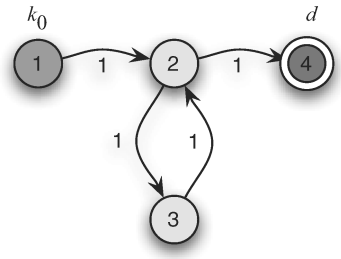

Figure 1: First network used in our simulations. The immediate costs are indicated on the arcs.

For instance, equations 7.2 and 7.3 provide recurrence relations that can be solved in a forward and a backward pass since the predecessors of node k come before k, while the successors of k come after k, once the nodes have been reordered. This leads to an algorithm closely related in spirit to dynamic programming or the forward and backward algorithm in hidden Markov models.

8 Simulation Results

We illustrate the procedures for solving the randomized shortest-path problem on three simple networks. For all networks, the algorithm detailed in section 4 (see algorithm 1) was iterated until convergence of the transition probabilities. We also used Akamatsu's model, as detailed in section 5 (see algorithm 2), which provides exactly the same results, as expected.

8.1 First Experiment. Our first experiment is performed on the simple network shown in Figure 1. The network is composed of only four nodes connected by arcs of different weights, representing costs. For this simple network, we easily observe that the average time needed to reach destination node 4 can be made arbitrarily large by increasing the probability of jumping from node 2 to node 3.

Figure 2 displays the average cost to reach destination node 4 when starting from initial node 1, in terms of global entropy H spread in the network. We clearly observe the increase in average cost when the entropy is increased.

For illustration purposes, Figure 3 shows the resulting Markov chains after convergence of the algorithm, including the transition probabilities ($p_{kk'}$) and the average cost ($v(1)$), for four different values of the global entropy H ($= 0.5, 1.0, 1.5, 2.0$).

8.2 Second Experiment. The second experiment is performed on the network shown in Figure 4. It is composed of eight nodes connected by edges of different weights, representing costs.

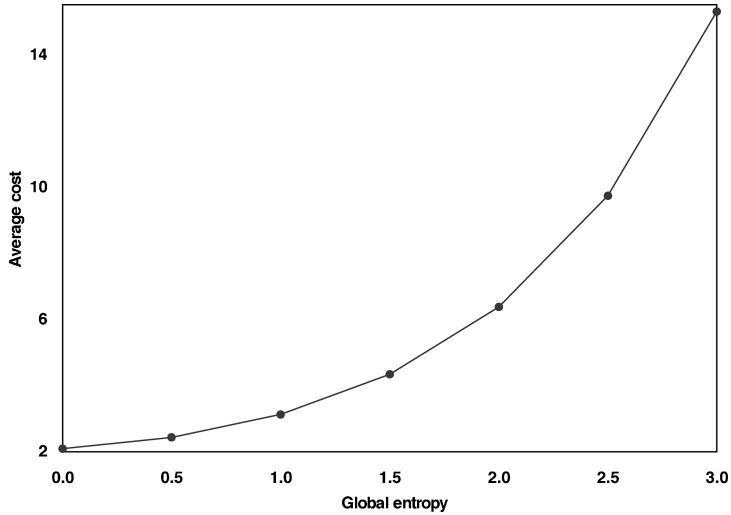

Figure 2: Average cost, $v(1)$, to reach destination node 4 when starting from initial node 1, in terms of global entropy H spread in the network.

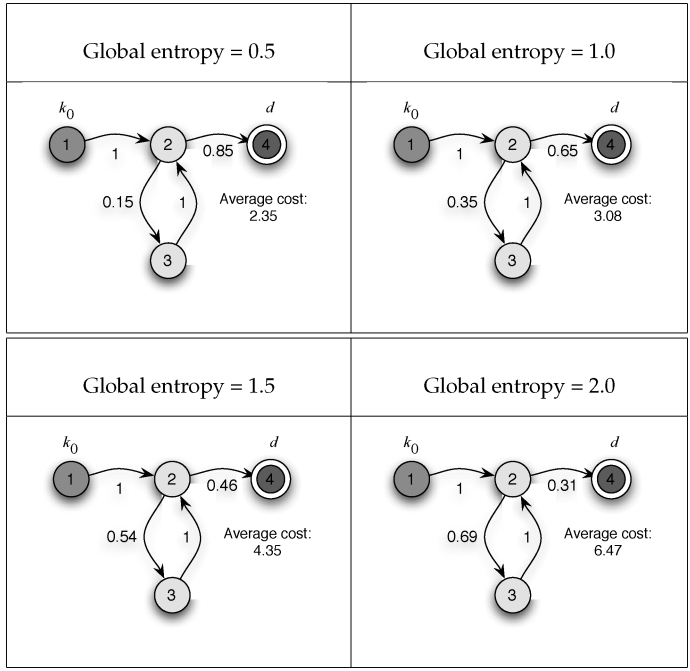

Figure 3: Resulting Markov chains together with the transition probabilities and the average cost, for four different values of the global entropy.

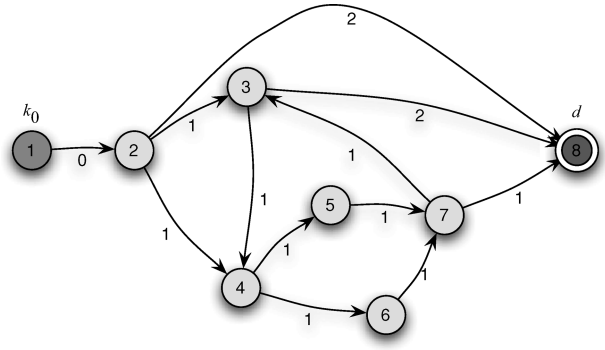

Figure 4: Second network used in our simulations. The immediate costs are indicated on the arcs.

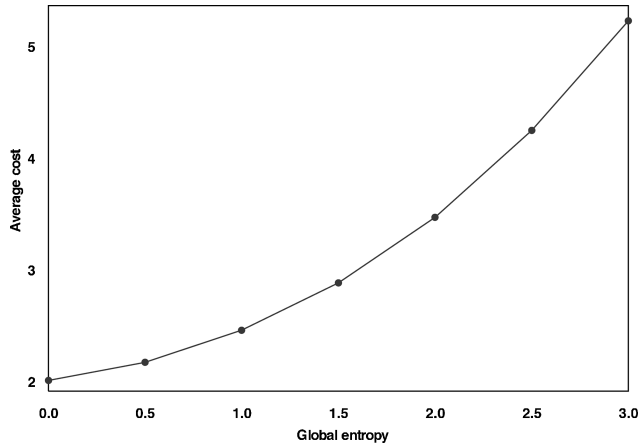

Figure 5: Average cost, $v(1)$, to reach destination node 8 when starting from initial node 1, in terms of global entropy H spread in the network.

As for the previous experiment, we display the average cost to reach destination node 8 when starting from initial node 1, in function of global entropy H spread in the network in Figure 5.

Figure 6 shows the resulting Markov chains after convergence of the algorithm for four different values of the global entropy.

8.3 Third Experiment. The third experiment is performed on a 40×30 grid. It aims to reach a goal node, located in the lower right corner, from an initial node, located in the upper left corner. An obstacle is placed on the grid so that the agents have to walk around in order to avoid it. The agent is allowed to move to a neighboring node, and a cost of 1 unit is incurred at each move.

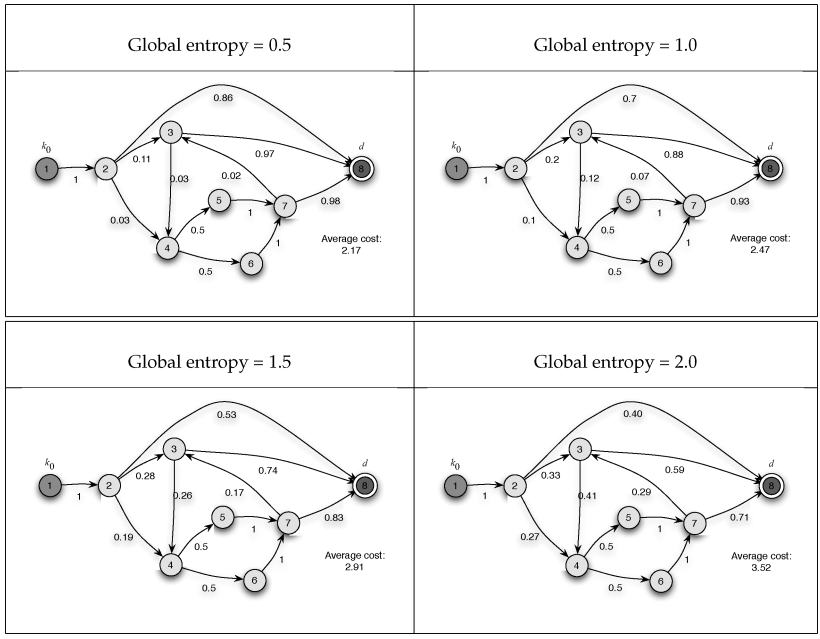

Figure 6: Resulting Markov chains together with the transition probabilities and the average cost for four values of the global entropy.

The resulting expected number of passages through every cell of the grid for three values of θ, provided by equation 5.17, is shown in Figure 7.

9 Conclusion

This work presented a preliminary study of the randomized shortest-path problem as well as two procedures for solving it.

The first one is similar to the value-iteration method for solving Markov decision processes and is iterative. Its main drawback is that it is computationally demanding since it relies on iterative algorithms and necessitates the solution of two linear systems of equations at each iteration. On the other hand, it exploits the sparseness of the network and relies on an iterative scheme, which can be useful when continual adaptation is needed, for instance, in changing environments.

The second procedure was originally introduced by Akamatsu in the framework of transportation networks. Based on Akamatsu's ideas and by revisiting the problem from a statistical physics perspective, we show that randomized shortest-path problems can be computed efficiently by solving a simple linear system of equations. This still is not very efficient in comparison with state-of-the-art algorithms solving

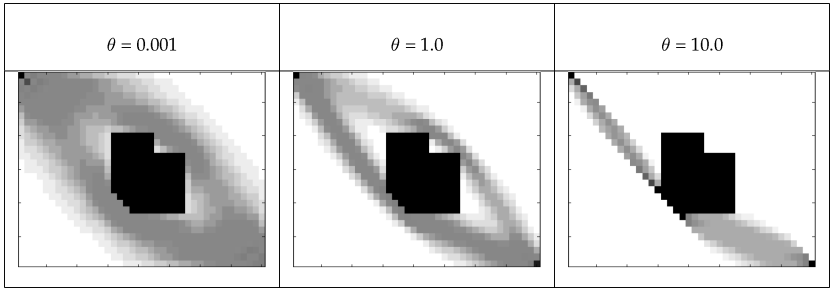

Figure 7: Expected number of passages through every cell of the grid, for three values of the parameter θ, and thus decreasing entropies. The agents, starting from the upper left corner, have to reach the lower right corner. An obstacle is placed in the center of the grid so that the agents have to walk around in order to avoid it.

single-source, single-destination, shortest-path problems, since solving a system of linear equations is $O(n^3)$, where n is the number of unknowns. However, for some sparse graphs having a special structure, this method could eventually prove useful; this will be investigated in further work.

Further work will be devoted to the analysis of the algorithm and the design of other procedures, differing in the definition of the global entropy quantifying the randomness in the network. We also plan to tackle Markov decision processes, as well as multiple-sources multiple-destinations problems, with this approach. Furthermore, since submission of this letter, we have exploited the randomized shortest-path distance as a dissimilarity measure between nodes for nodes clustering or betweenness computation (Yen et al., 2008). A covariance measure between nodes could also be defined within the same sum-over-paths framework: two nodes would be considered as correlated if they often co-occur on the same path.

A further application would be to design randomized edit distances or kernel-based sequence alignment procedures accounting for all editing and alignment paths. Yet another application would be to extend some of the results to the so-called semiring framework (Carre, 1979; Gondran & Minoux, 1984; Mohri, 2002). Finally, we also plan to investigate the links between our proposed models and the recent work of Nesterov (2007).

Appendix: Proof of the Main Results

A.1 Computation of the Expected Cost-to-Go in Terms of the Fundamental Matrix. Let us start from equation 2.3, which is restated here:

$$\mathbf{v} = \mathbf{diag}(\mathbf{P}\mathbf{C}^{\mathrm{T}}) + \mathbf{P}\mathbf{v}.$$

Thus, we have

$$\begin{bmatrix} \widetilde{\mathbf{v}} \\ 0 \end{bmatrix} = \mathbf{diag}\left(\begin{bmatrix} \mathbf{Q} & \mathbf{r} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{D}^{\mathrm{T}} & \boldsymbol{\infty} \\ \mathbf{s}^{\mathrm{T}} & 0 \end{bmatrix} \right) + \begin{bmatrix} \mathbf{Q} & \mathbf{r} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{v}} \\ 0 \end{bmatrix}, \tag{A.1}$$

from which we easily deduce

$$\widetilde{\mathbf{v}} = \mathbf{diag}(\mathbf{Q}\mathbf{D}^{\mathrm{T}} + \mathbf{r}\mathbf{s}^{\mathrm{T}}) + \mathbf{Q}\widetilde{\mathbf{v}}. \tag{A.2}$$

Isolating $\widetilde{\mathbf{v}}$ provides us with the required result:

$$\begin{aligned} \widetilde{\mathbf{v}} &= (\mathbf{I} - \mathbf{Q})^{-1}\, \mathbf{diag}(\mathbf{Q}\mathbf{D}^{\mathrm{T}} + \mathbf{r}\mathbf{s}^{\mathrm{T}}) \\ &= \mathbf{N}\, \mathbf{diag}(\mathbf{Q}\mathbf{D}^{\mathrm{T}} + \mathbf{r}\mathbf{s}^{\mathrm{T}}), \end{aligned} \tag{A.3}$$

where $\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1}$. This equation expresses the expected cost-to-go in terms of the fundamental matrix $\mathbf{N}$, the transition probabilities, and the cost matrix.

A.2 Determination of the Optimal Policy. The goal here is to determine the set of transition probabilities $\pi \equiv \{p_{kk'}; k = 1, 2, \ldots, (n-1); k' = 2, \ldots, n\}$ that minimizes the expected cost, when starting from state 1 and subject to the entropy constraint (see equation 2.10). We therefore introduce the following Lagrange function, taking all the constraints into account, namely the equations computing the mean expected cost at each node (see equation 2.2; parameters λ_k), the sum-to-zero constraints for the transition probabilities at each node (parameters μ_k), and the entropy constraint (see equation 2.10, parameter η),

$$\begin{aligned} \mathcal{L} = v(1) &+ \sum_{k \neq n} \lambda_k \left[v(k) - \sum_{k' \in S(k)} p_{kk'} \left(c_{kk'} + v(k') \right) \right] \\ &+ \lambda_n [v(n) - 0] + \sum_{k \neq n} \mu_k \left[\sum_{k' \in S(k)} p_{kk'} - 1 \right] \\ &+ \eta \left[\sum_{k \neq n} u_k \sum_{k' \in S(k)} p_{kk'} \log p_{kk'} + H_0 \right]. \end{aligned} \tag{A.4}$$

Strictly speaking, a constraint guaranteeing the positivity of the transition probabilities should be introduced as well, but this is not necessary since the resulting transition probabilities will automatically be positive.

Differentiating this Lagrange function in terms of the transition probabilities, $\partial \mathcal{L}/\partial p_{ll'}$, and equating to zero gives

$$-\lambda_l(c_{ll'} + v(l^{'})) + \mu_l + \eta\, u_l(\log p_{ll'} + 1) - \eta \sum_{k \neq n}(\partial u_k/\partial p_{ll'})h_k = 0, \quad \text{(A.5)}$$

where $h_k = -\sum_{k' \in S(k)} p_{kk'} \log p_{kk'}$. Extracting $\log p_{ll'}$ from this equation provides

$$\eta\, u_l \log p_{ll'} = -\mu_l - \eta\, u_l + \lambda_l(c_{ll'} + v(l')) + \eta \sum_{k \neq n}(\partial u_k/\partial p_{ll'})h_k. \quad \text{(A.6)}$$

Exponentiating this last equation yields

$$p_{ll'} = \exp\left[\frac{-\mu_l - \eta\, u_l}{\eta\, u_l}\right] \exp\left[\frac{\lambda_l(c_{ll'} + v(l')) + \eta \sum_{k \neq n}(\partial u_k/\partial p_{ll'})h_k}{\eta\, u_l}\right]. \quad \text{(A.7)}$$

Summing the equation over $l' \in S(l)$ and observing that the probabilities sum to one allows us to compute the first factor of the right-hand side:

$$\exp\left[\frac{-\mu_l - \eta\, u_l}{\eta\, u_l}\right] = \left[\sum_{l' \in S(l)} \exp\left[\frac{\lambda_l(c_{ll'} + v(l')) + \eta \sum_{k \neq n}(\partial u_k/\partial p_{ll'})h_k}{\eta\, u_l}\right]\right]^{-1}. \quad \text{(A.8)}$$

By replacing equation A.8 in equation A.7 and defining $\theta_k = -\lambda_k$, we finally obtain

$$p_{kk'} = \frac{\exp\left[-\frac{\theta_k}{\eta\, u_k}(c_{kk'} + v(k')) + \frac{1}{u_k}\sum_{l \neq n}(\partial u_l/\partial p_{kk'})h_l\right]}{\sum_{l' \in S(k)} \exp\left[-\frac{\theta_k}{\eta\, u_k}(c_{kl'} + v(l')) + \frac{1}{u_k}\sum_{l \neq n}(\partial u_l/\partial p_{kl'})h_l\right]}. \quad \text{(A.9)}$$

Now, differentiating $\mathcal{L}$ in terms of the expected costs, $\partial \mathcal{L}/\partial v(l)$, $l \neq 1$, and equating to zero allows us to compute the Lagrange multipliers λ_l,

$$\lambda_l = \sum_{k \in P(l)} p_{kl}\lambda_k, \quad \text{for } l \neq 1, \quad \text{(A.10)}$$

$P(l)$ being the set of nodes from which node l is accessible in one step (the predecessors of l). For the initial state, we obtain (recall that we assumed in

section 2 that the initial state has no predecessor)

$$\lambda_1 = -1. \tag{A.11}$$

Defining $\theta_k = -\lambda_k$ provides

$$\begin{cases} \theta_l = \sum_{k \in P(l)} p_{kl}\, \theta_k, & \text{for } l \neq 1 \\ \theta_1 = 1, & \text{for initial state 1} \end{cases}. \tag{A.12}$$

Rewriting equation A.12 in matrix form gives $\boldsymbol{\theta} = \mathbf{P}^{\mathrm{T}}\boldsymbol{\theta} + \mathbf{e}_1$. Therefore, when considering only the transient states, $\widetilde{\boldsymbol{\theta}} = \mathbf{Q}^{\mathrm{T}}\widetilde{\boldsymbol{\theta}} + \widetilde{\mathbf{e}}_1$. In other words, we have

$$\widetilde{\boldsymbol{\theta}} = (\mathbf{I} - \mathbf{Q}^{\mathrm{T}})^{-1}\widetilde{\mathbf{e}}_1. \tag{A.13}$$

By comparing equation A.13 with equation 2.5, we easily observe that the Lagrange parameters vector $\widetilde{\boldsymbol{\theta}}$ simply corresponds to the expected number of visits to each transient state before reaching the absorbing state—in other words, $\widetilde{\boldsymbol{\theta}} = \mathbf{n}$ in equation A.9.

Finally, expressing the fact that the policy has a fixed entropy,

$$-\sum_{k \neq n} u_k \sum_{k' \in S(k)} p_{kk'} \log p_{kk'} = H_0, \tag{A.14}$$

allows us to compute the value of η in terms of H_0.

A.3 Computation of $\sum_{l \neq n}(\partial n_l / \partial p_{kk'})h_l$. Let us first compute $\sum_{l \neq n}(\partial n_l / \partial p_{kk'})h_l$ for $k, k' \neq n$. We have

$$\begin{aligned} \sum_{l \neq n}(\partial n_l / \partial p_{kk'})h_l &= \frac{\partial \mathbf{n}^{\mathrm{T}}}{\partial p_{kk'}}\mathbf{h} \\ &= \frac{\partial (\mathbf{N}^{\mathrm{T}}\widetilde{\mathbf{e}}_1)^{\mathrm{T}}}{\partial p_{kk'}}\mathbf{h} \\ &= \widetilde{\mathbf{e}}_1^{\mathrm{T}}\frac{\partial \mathbf{N}}{\partial p_{kk'}}\mathbf{h}, \end{aligned} \tag{A.15}$$

where we used $\mathbf{n} = \mathbf{N}^{\mathrm{T}}\widetilde{\mathbf{e}}_1$ (see equation 2.5).

But since $\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1}$,

$$\begin{aligned} \frac{\partial \mathbf{N}}{\partial p_{kk'}} &= \frac{\partial (\mathbf{I} - \mathbf{Q})^{-1}}{\partial p_{kk'}} \\ &= -(\mathbf{I} - \mathbf{Q})^{-1}\frac{\partial (\mathbf{I} - \mathbf{Q})}{\partial p_{kk'}}(\mathbf{I} - \mathbf{Q})^{-1} \end{aligned}$$

$$= \mathbf{N}\frac{\partial \mathbf{Q}}{\partial q_{kk'}}\mathbf{N}$$
$$= \mathbf{N}\widetilde{\mathbf{e}}_k\widetilde{\mathbf{e}}_{k'}^{\mathrm{T}}\mathbf{N}, \tag{A.16}$$

where we used the standard formula for matrix differentiation, $d\mathbf{M}^{-1} = -\mathbf{M}^{-1}(d\mathbf{M})\mathbf{M}^{-1}$ (see, e.g., Harville, 1997). Replacing equation A.16 in A.15 provides

$$\begin{aligned}\sum_{l\neq n}(\partial n_l/\partial p_{kk'})h_l &= \widetilde{\mathbf{e}}_1^{\mathrm{T}}\mathbf{N}\widetilde{\mathbf{e}}_k\widetilde{\mathbf{e}}_{k'}^{\mathrm{T}}\mathbf{N}\mathbf{h}\\ &= \mathbf{n}^{\mathrm{T}}\widetilde{\mathbf{e}}_k\widetilde{\mathbf{e}}_{k'}^{\mathrm{T}}\mathbf{N}\mathbf{h}\\ &= n_k\widetilde{\mathbf{e}}_{k'}^{\mathrm{T}}\mathbf{N}\mathbf{h}\\ &= n_k\sum_{l\neq n}n_{k'l}h_l,\end{aligned} \tag{A.17}$$

which is the required result.

Now, when $k' = n$ (absorbing node), since the n_l do not depend explicitely on the p_{kn} (n_l only depends on $\mathbf{Q}$), $\partial n_l/\partial p_{kn} = 0$ and thus $\sum_{l\neq n}(\partial n_l/\partial p_{kn})h_l = 0$.

A.4 Computation of the Derivatives of the Partition Function $\mathcal{Z}$. Notice first that in this section, we assume $k \neq n$ and $n \neq 1$ (n is the destination, absorbing, state). Remember that the partition function is $\mathcal{Z} = [(\mathbf{I}-\mathbf{W})^{-1}]_{1n}$. First, we have to compute $\overline{E}$:

$$\overline{E} = \frac{\partial(-\log\mathcal{Z})}{\partial\theta} = -\frac{\partial_\theta\left[\mathbf{e}_1^{\mathrm{T}}(\mathbf{I}-\mathbf{W})^{-1}\mathbf{e}_n\right]}{\mathbf{e}_1^{\mathrm{T}}(\mathbf{I}-\mathbf{W})^{-1}\mathbf{e}_n} = -\frac{\mathbf{e}_1^{\mathrm{T}}\partial_\theta(\mathbf{I}-\mathbf{W})^{-1}\mathbf{e}_n}{\mathbf{e}_1^{\mathrm{T}}(\mathbf{I}-\mathbf{W})^{-1}\mathbf{e}_n}. \tag{A.18}$$

Let us compute $\partial_\theta(\mathbf{I}-\mathbf{W})^{-1}$. By setting $\mathbf{Z} = (\mathbf{I}-\mathbf{W})^{-1}$ and denoting element i, j of $\mathbf{Z}$ by z_{ij}, we obtain

$$\begin{aligned}\partial_\theta(\mathbf{I}-\mathbf{W})^{-1} &= -\mathbf{Z}(\partial_\theta(\mathbf{I}-\mathbf{W}))\mathbf{Z}\\ &= \mathbf{Z}(\partial_\theta\mathbf{W})\mathbf{Z}\\ &= \mathbf{Z}\mathbf{W}'_\theta\mathbf{Z},\end{aligned} \tag{A.19}$$

where the matrix $\mathbf{W}'_\theta = \partial\mathbf{W}/\partial\theta$ contains the elements $[\mathbf{W}'_\theta]_{ij} = -c_{ij}\exp[-\theta c_{ij}]$. Therefore, the expected cost needed for reaching state n

from state 1, $\overline{E}$, is

$$\overline{E} = -\frac{\mathbf{e}_1^{\mathrm{T}}\mathbf{Z}\mathbf{W}'_\theta\mathbf{Z}\mathbf{e}_n}{\mathbf{e}_1^{\mathrm{T}}\mathbf{Z}\mathbf{e}_n} = -\frac{\mathbf{e}_1^{\mathrm{T}}\mathbf{Z}\mathbf{W}'_\theta\mathbf{Z}\mathbf{e}_n}{z_{1n}}. \tag{A.20}$$

We now turn to the computation of $\overline{\eta}_{kk'}$. We easily find

$$\begin{aligned}
\overline{\eta}_{kk'} &= \frac{\partial F}{\partial c_{kk'}} = -\frac{1}{\theta}\frac{\partial \log(\mathcal{Z})}{\partial c_{kk'}} \\
&= -\frac{1}{\theta}\frac{\partial_{c_{kk'}}\left[\mathbf{e}_1^{\mathrm{T}}(\mathbf{I}-\mathbf{W})^{-1}\mathbf{e}_n\right]}{\mathbf{e}_1^{\mathrm{T}}(\mathbf{I}-\mathbf{W})^{-1}\mathbf{e}_n} \\
&= -\frac{1}{\theta}\frac{\mathbf{e}_1^{\mathrm{T}}\partial_{c_{kk'}}(\mathbf{I}-\mathbf{W})^{-1}\mathbf{e}_n}{\mathbf{e}_1^{\mathrm{T}}(\mathbf{I}-\mathbf{W})^{-1}\mathbf{e}_n} \\
&= -\frac{1}{\theta}\frac{\mathbf{e}_1^{\mathrm{T}}\partial_{c_{kk'}}\mathbf{Z}\mathbf{e}_n}{\mathbf{e}_1^{\mathrm{T}}\mathbf{Z}\mathbf{e}_n} = -\frac{1}{\theta}\frac{\mathbf{e}_1^{\mathrm{T}}(\partial_{c_{kk'}}\mathbf{Z})\mathbf{e}_n}{z_{1n}}.
\end{aligned} \tag{A.21}$$

Let us compute $\partial_{c_{kk'}}\mathbf{Z} = \partial_{c_{kk'}}(\mathbf{I}-\mathbf{W})^{-1}$:

$$\begin{aligned}
\partial_{c_{kk'}}(\mathbf{I}-\mathbf{W})^{-1} &= -\mathbf{Z}(\partial_{c_{kk'}}(\mathbf{I}-\mathbf{W}))\mathbf{Z} \\
&= \mathbf{Z}(\partial_{c_{kk'}}\mathbf{W})\mathbf{Z} \\
&= -\theta\ \exp\left[-\theta c_{kk'}\right]\mathbf{Z}\mathbf{e}_k\mathbf{e}_{k'}^{\mathrm{T}}\mathbf{Z}.
\end{aligned} \tag{A.22}$$

Thus, $\overline{\eta}_{kk'}$ is given by

$$\overline{\eta}_{kk'} = \exp\left[-\theta c_{kk'}\right]\frac{\mathbf{e}_1^{\mathrm{T}}\mathbf{Z}\mathbf{e}_k\mathbf{e}_{k'}^{\mathrm{T}}\mathbf{Z}\mathbf{e}_n}{z_{1n}} = \frac{z_{1k}z_{k'n}\exp\left[-\theta c_{kk'}\right]}{z_{1n}}.$$

The expected number of passages through state k' is given by

$$\overline{n}_{k'} = \frac{z_{k'n}}{z_{1n}}\sum_{k=1}^{n} z_{1k}\exp\left[-\theta c_{kk'}\right] = \frac{z_{1k'}z_{k'n}}{z_{1n}}, \tag{A.23}$$

where we used $z_{1k'} = \sum_{k=1}^{n} z_{1k}\exp\left[-\theta c_{kk'}\right]$, which directly follows from equation 7.2.

The transition probabilities are

$$p_{kk'} = \frac{\overline{\eta}_{kk'}}{\overline{n}_k} = \frac{z_{k'n}}{z_{kn}}\exp\left[-\theta c_{kk'}\right], \tag{A.24}$$

which corresponds exactly to the result of Akamatsu, derived in a completely different way (see Akamatsu, 1996).

Finally, the formula for the entropy has already been derived at the end of section 5.1; see equations 5.8 and 5.9.

Acknowledgments

We thank the anonymous reviewers for their interesting remarks and suggestions that allowed us to improve significantly the quality of the letter. We also thank Masashi Shimbo for pointing out the relationships with conditional random fields and François Glineur for his useful insights on optimization theory.

Part of this work has been funded by projects with the Région wallonne and the Région de Bruxelles-Capitale.

References

Achbany, Y., Fouss, F., Yen, L., Pirotte, A., & Saerens, M. (2006). Optimal tuning of continual exploration in reinforcement learning. In *Proceedings of the 16th International Conference on Artificial Neural Networks (ICANN 06)* (pp. 734–749). Berlin: Springer-Verlag.

Achbany, Y., Fouss, F., Yen, L., Pirotte, A., & Saerens, M. (2008). Tuning continual exploration in reinforcement learning: An optimality property of the Boltzmann strategy. *Neurocomputing*, *71*, 2507–2520.

Adelson-Velsky, G. M., Arlazarov, V. L., & Donskoy, M. V. (1988). *Algorithms for games*. Berlin: Springer-Verlag.

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

Akamatsu, T. (1997). Decomposition of path choice entropy in general transport networks. *Transportation Science*, *31*(4), 349–362.

Bather, J. (2000). *Decision theory: An introduction to dynamic programming and sequential decisions*. Hoboken, NJ: Wiley.

Bertsekas, D. P. (1998). *Network optimization: Continuous and discrete models*. Belmont, MA: Athena Scientific.

Bertsekas, D. P. (1999). *Nonlinear programming* (2nd ed.). Belmont, MA: Athena Scientific.

Bertsekas, D. P. (2000). *Dynamic programming and optimal control*. Belmont, MA: Athena Scientific.

Bertsekas, D. P., & Gallager, R. (1992). *Data networks* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Bertsekas, D. P., & Tsitsiklis, J. (1996). *Neuro-dynamic programming*. Belmont, MA: Athena Scientific.

Boyd, S., Diaconis, P., & Xiao, L. (2004). Fastest mixing Markov chain on a graph. *SIAM Review*, *46*, 667–689.

Bronson, R. (1989). *Matrix operations*. New York: McGraw-Hill.

Carre, B. (1979). *Graphs and networks*. New York: Oxford University Press.

Christofides, N. (1975). *Graph theory: An algorithmic approach*. Orlando, FL: Academic Press.

Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory* (2nd ed.). Hoboken, NJ: Wiley.

Davis, T. A. (2006). *Direct methods for sparse linear systems*. Philadelphia, PA: SIAM.

Delvenne, J.-C. (2005). *Pagerank, entropy and free energy*. Manuscript submitted for publication.

Dial, R. (1971). A probabilistic multipath assignment model that obviates path enumeration. *Transportation Research*, *5*, 83–111.

Dorigo, M., & Stutzle, T. (2004). *Ant colony optimization*. Cambridge, MA: MIT Press.

Durbin, R., Eddy, S. R., Krogh, A., & Mitchison, G. (1998). *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge: Cambridge University Press.

Ekroot, L., & Cover, T. (1993). The entropy of Markov trajectories. *IEEE Transactions on Information Theory*, *39*(4), 1418–1421.

Feynman, R. (1948). Space-time approach to non-relativistic quantum mechanics. *Reviews of Modern Physics*, *20*(2), 367–387.

Fouss, F., Pirotte, A., Renders, J.-M., & Saerens, M. (2007). Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *19*(3), 355–369.

Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated planning: Theory and practice*. San Francisco: Morgan Kaufmann.

Girardin, V. (2004). Entropy minimization for markov and semi-Markov processes. *Methodology and Computing in Applied Probability*, *6*, 109–127.

Girardin, V., & Limnios, N. (2004). Entropy rate and maximum entropy methods for countable semi-Markov chains. *Communications in Statistics*, *33*(3), 609–622.

Gondran, M., & Minoux, M. (1984). *Graphs and algorithms*. Hoboken, NJ: Wiley.

Gusfield, D. (1997). *Algorithms on strings, trees and sequences*. Cambridge: Cambridge University Press.

Harville, D. A. (1997). *Matrix algebra from a statistician's perspective*. Berlin: Springer-Verlag.

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

Jelinek, F. (1997). *Statistical methods for speech recognition*. Cambridge, MA: MIT Press.

Jungnickel, D. (2004). *Graphs, networks and algorithms* (2nd ed.). Berlin: Springer-Verlag.

Kapur, J. N., & Kesavan, H. K. (1992). *Entropy optimization principles with applications*. Orlando, FL: Academic Press.

Kemeny, J. G., & Snell, J. L. (1976). *Finite Markov chains* (rev. ed.). Berlin: Springer-Verlag.

Klein, D. J., & Randic, M. (1993). Resistance distance. *Journal of Mathematical Chemistry*, *12*, 81–95.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML)* (pp. 282–289). San Francisco: Morgan Kaufmann.

LaValle, S. (2006). *Planning algorithms*. Cambridge: Cambridge University Press.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning (ICML-94)* (pp. 157–163). San Francisco: Morgan Kaufmann.

Mohri, M. (2002). Semiring frameworks and algorithms for shortest-distance problems. *Journal of Automata, Languages and Combinatorics*, *7*(3), 321–350.

Motwani, R., & Raghavan, P. (1995). *Randomized algorithms*. Cambridge: Cambridge University Press.

Nadler, B., Lafon, S., Coifman, R., & Kevrekidis, I. (2006). Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators. In Y. Weiss, B. Schölkopf, & J. Platt (Eds.), *Advances in neural information processing systems, 18* (pp. 955–962). Cambridge, MA: MIT Press.

Nesterov, Y. (2007). Characteristic functions of directed graphs and applications to stochastic equilibrium problems. *Optimization and Engineering*, *8*(2), 193–214.

Norris, J. (1997). *Markov chains*. Cambridge: Cambridge University Press.

Osborne, M. J. (2004). *An introduction to game theory*. New York: Oxford University Press.

Powell, W. (2007).*Approximate dynamic programming*. Hoboken, NJ: Wiley.

Puterman, M. (1994). *Markov decision processes: Discrete stochastic programming*. Hoboken, NJ: Wiley.

Qiu, H., & Hancock, E. R. (2007). Clustering and embedding using commute times. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(11), 1873–1890.

Rabiner, L., & Juang, B.-H. (1993). *Fundamentals of speech recognition*. Upper Saddle River, NJ: Prentice Hall.

Raiffa, H. (1970). *Decision analysis*. Reading, MA: Addison-Wesley.

Reichl, L. (1998). *A modern course in statistical physics* (2nd ed.). Hoboken, NJ: Wiley.

Rue, H., & Held, L. (2005). *Gaussian Markov random fields*. London: Chapman & Hall.

Saerens, M., Fouss, F., Yen, L., & Dupont, P. (2004). The principal components analysis of a graph, and its relationships to spectral clustering. In *Proceedings of the 15th European Conference on Machine Learning (ECML 2004)* (pp. 371–383). Berlin: Springer-Verlag.

Schrödinger, E. (1952). *Statistical thermodynamics* (2nd ed.). Cambridge: Cambridge University Press.

Sedgewick, R. (1990). *Algorithms in C*. Reading, MA: Addison-Wesley.

Shannon, C. (1948). A mathematical theory of communication. *Bell System Technical Journal*, *27*, 379–423, 623–656.

Sun, J., Boyd, S., Xiao, L., & Diaconis, P. (2006). The fastest mixing Markov process on a graph and a connection to a maximum variance unfolding problem. *SIAM Review, 48*, 681–699.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

Tahbaz, A., & Jadbabaie, A. (2006). A one-parameter family of distributed consensus algorithms with boundary: From shortest paths to mean hitting times. In *Proceedings of IEEE Conference on Decision and Control* (pp. 4664–4669). Piscataway, NJ: IEEE Press.

Taylor, H. M., & Karlin, S. (1998). *An introduction to stochastic modeling* (3th ed.). Orlando, FL: Academic Press.

Todorov, E. (2006). Linearly-solvable Markov decision problems. In B. Schölkopf, J. Platt, & T. Hoffman (Eds.), *Advances in neural information processing systems, 19* (pp. 1369–1375). Cambridge, MA: MIT Press.

Tomlin, J. (2003). A new paradigm for ranking pages on the World Wide Web. In *Proceedings of the International World Wide Web Conference (WWW2003)* (pp. 350–355). New York: ACM Press.

Wasserman, S., & Faust, K. (1994). *Social network analysis: Methods and applications*. Cambridge: Cambridge University Press.

Yen, L., Mantrach, A., Shimbo, M., & Saerens, M. (2008). 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)* (pp. 785–793). New York: ACM Press.

Received November 7, 2007; accepted January 7, 2009.