Randomized Shortest-Path Problems: Two related models

Marco Saerens\textsuperscript{1,}\textsuperscript{*}, Youssef Achbany\textsuperscript{1}, François Fouss\textsuperscript{1,2} & Luh Yen\textsuperscript{1}

\textsuperscript{1}Information Systems Unit (ISYS) and Machine Learning Group (MLG) Université catholique de Louvain (UCL), Belgium
\{marco.saerens, youssef.achbany, luh.yen\}@uclouvain.be

\textsuperscript{2}Management Sciences Department Facultés Universitaires Catholiques de Mons (FUCaM), Belgium
francois.fouss@fucam.ac.be

December 23, 2008

Abstract

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, to avoid congestion, or to avoid complete predictability of your routing strategy. In other words, you want to introduce some randomness/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 allowing to compute the optimal randomized policy – minimizing the expected routing cost – are derived. Iterating these necessary conditions, reminiscent of Bellman’s value iteration equations, allows to compute 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 (Akamatsu (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 to easily derive 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

\*Marco Saerens is also a Research Fellow of the IRIDIA Laboratory, Université Libre de Bruxelles.
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, artificial intelligence, to name a few. Many extensions of the basic shortest-path algorithms have been proposed, still extending their application range (see for instance 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 and Juang (1993)), sequence alignment (Durbin et al. (1998); Gusfield (1997)), Markov decision processes (Bather (2000); Puterman (1994)), routing in data networks (Bertsekas and Gallager (1992)), social network analysis (betweenness centrality; see Wasserman and Faust (1994)), game playing (minimax algorithms; see, e.g., Adelson-Velsky et al. (1988)), planning (Ghallab et al. (2004); LaValle (2006)), neurocomputing (Bertsekas and Tsitsiklis (1996)), reinforcement learning (Sutton and 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 will be called 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 just 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 the present paper, 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 (non-stationary), 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 paper, 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 intro-
duces unpredictability and therefore renders interception more difficult.
Randomization has proven useful for exactly this reason in game theory.

- In the case where there are multiple destination nodes (or goals), intro-
ducing randomness allows to perform some load balancing, by exploit-
ing the goal nodes in parallel.

- Randomization also allows to spread the traffic on multiple paths, there-
fore reducing the danger of congestion. Indeed, by following a random-
ized strategy, the goods are routed along multiple different paths and are
therefore spread over the network.

- One may want to use a dissimilarity measure between two nodes that
does not only account for the shortest path, but also for all the other
paths, with longer paths being penalized with respect to short ones; there-
fore considering that nodes connected by many, short, paths are closer
than nodes connected by, for instance, only one short path (as in Fouss
et al. (2007)).

- In some applications areas, such as sequence alignment, computing a
similarity measure accounting for all paths could eventually provide bet-
ter results than relying on the best path. This is, of course, 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 (for instance an entropy constraint) is
added in order to obtain optimal randomized policies (also called stochastic policies),
and therefore continual exploration.

Mathematically speaking, 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 to measure the randomness associated to
a given node by the (Shannon) entropy (see for instance Cover and Thomas
(2006); Kapur and Kesavan (1992); Shannon (1948)) of the probability distri-
bution 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 en-
tropy 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 the related work), 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 et al. (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 and 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 the present work, the global degree of randomness associated to the whole network is quantified by a weighted sum of the individual entropies associated to 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 for instance Osborne (2004)) and Markov games (Littman (1994)), were not yet 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 probability 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 probability 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 the random utility theory. The present work therefore provides a new interpretation for Akamatsu’s model, as well as a new perspective, based on statistical physics, which allows to derive the main results in a unified way.

Let us finally mention some interesting papers that are 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 the present work, while 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). In this paper, the authors provide formulas allowing to compute the entropy needed to reach the destination node. Tahbaz and Jadbabaie (2006) introduced a one-parameter family of algorithms that, as our algorithm, recover both the Bellman-Ford procedure for finding shortest paths as well as the iterative algorithm for computing the average fist-passage time. However, it was based on heuristic grounds and 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 et al. (2004) design a Markov chain that has fastest mixing properties and Sun et al. (2006) discuss its continuous-time counterpart. In a completely different framework, uninformed random walks, based on maximizing the 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 (non-absorbing) Markov or semi-Markov chains in a maximum entropy framework (see for instance Girardin (2004); Girardin and Limnios (2004), and the references therein). This work is based on ergodic Markov chains while the present paper deals with absorbing Markov chains and randomized shortest-path problems.

1.2 Contributions and organization of the paper

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 as a first step towards 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 paper provides a new interpretation for Akamatsu’s model: it is optimal in the sense that it minimizes expected cost for a fixed entropy spread in the network.

- Akamatsu’s model (Akamatsu (1996)) is revisited in a unified statistical physics formalism, which allows to easily compute all the quantities of interest, in particular the policy. It involves the introduction of 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 was not noticed by Akamatsu. 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 or 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-persons 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 a 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 et al. (2008) as an application of the models introduced in this paper. This fact can be exploited in data-mining applications such as recommender systems (see for instance Saerens et al. (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 and Randic (1993); Nadler et al. (2006); Qiu and 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 as propositions for further work. Moreover, we have to stress the fact that the present work is not focused on any specific application area; rather, its main purpose is to study RSP problems per se and to propose some general techniques to tackle them.
What concerns the organization of the paper, 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 into 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 is the conclusion.

2 Statement of the problem and notations

For the sake of simplicity, this paper will be focused on what is called “deterministic shortest-path problems”¹ as defined, for instance, by Bertsekas (2000). 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 neighbouring nodes (successors $S$) that can be reached from node $k$. These transition probabilities, defined on each node $k$, will be denoted as $p(k'|k) = p_{kk'}$ with $k' \in S(k)$. Furthermore, $P$ 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 $\infty$; in this case, the corresponding transition probability is set to zero, $p_{kk'} = 0$. The main difference between randomized and standard deterministic shortest-paths 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 for instance Osborne (2004)), computer sciences (Motwani and Raghavan (1995)), Markov games (Littman (1994)) or decision sciences (Raiffa (1970)).

Moreover, as Bertsekas (2000), we assume that 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 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

¹Notice that in this work paths may contain cycles. Paths containing cycles are also commonly called walks in the literature.
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 and Tsitsiklis (1996).

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

\[
v_\pi(k_0) = \mathbb{E}_\pi \left\{ \sum_{t=0}^{\infty} c_{s_t s_{t+1}} | s_0 = k_0 \right\}
\]

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 \( \pi \). The expectation is taken with respect to the transition probabilities associated to the nodes.

Thus, the main objective is to design a randomized policy minimizing the expected cost-to-go (Equation (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, \( \pi^* = \arg\min_\pi [v_\pi(k_0)] \) or, equivalently, an optimal transition-probabilities matrix \( P^* \) 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, \( \pi \), 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 (1) from any state \( k \) is closely related to the computation of the “average first-passage time” in the associated Markov chain (Kemeny and 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 for instance Taylor and 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

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

These equations can be iterated in order to find the expected costs; a closed-form solution will be derived later and necessitates the introduction 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:

\[
v = \text{diag}(PC^T) + Pv
\]
where \( \text{diag}(M) \) is a column vector containing the elements on the diagonal of matrix \( M \), \( v \) is a \( n \times 1 \) column vector containing the expected costs-to-go, \( v_s(k) \), for each of the \( (n-1) \) first states and a 0 as last \( (n\text{th}) \) element, while \( C \) is the matrix containing the immediate costs \( c_{kk'} \) (with \( c_{nn} = 0 \)).

From this Markov chain, one can also compute the expected number of visits to each state in the following way (Kemeny and 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 \( P \) can therefore be rewritten in partitioned form,

\[
P = \begin{bmatrix} Q & r \\ 0^T & 1 \end{bmatrix}
\]

where \( Q \) is the \((n-1) \times (n-1)\) substochastic matrix of transition probabilities among the transient states, \( r \) is a \((n-1) \times 1\) column vector representing the transition probabilities from transient states to the absorbing state \( n \) and \( 0 \) is an \((n-1) \times 1\) column vector full of 0’s. Now, 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 \( x(t) = (P^T)^t e_1 \) where \( x(t) \) 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 \( e_1 \) is a \( n \times 1 \) column vector containing a 1 as first element and 0’s otherwise. Focusing the analysis on the transient states only, the corresponding probability of finding the random walker in any transient state at time step \( t \) is \( \tilde{x}(t) = (Q^T)^t e_1 \), where \( \tilde{x}(t) \) and \( e_1 \) are \((n-1) \times 1\) column vectors obtained from \( x(t) \) and \( e_1 \) by removing the last \((n\text{th})\) element.

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

\[
n = \sum_{t=0}^{\infty} \tilde{x}(t) = \sum_{t=0}^{\infty} (Q^T)^t e_1 = (I - Q^T)^{-1} e_1 = N^T e_1 \tag{5}
\]

The matrix \( N = (I - Q)^{-1} \) is usually called the fundamental matrix of the Markov chain (Kemeny and Snell (1976); Norris (1997)). Its elements \( n_{ij} = [N]_{ij} \) 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 \( n \) contains the expected number of visits to each transient state, when starting from state 1.

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

\[
C = \begin{bmatrix} D & s \\ \infty^T & 0 \end{bmatrix},
\]

Equation (3) can be solved in terms of \( v \) and re-expressed as (see Appendix A)

\[
\tilde{v} = N \text{diag}(QD^T + rs^T) \tag{7}
\]

where \( \tilde{v} \) is a \((n-1) \times 1\) column vector containing the \((n-1)\) first elements of \( v \) (the \( n\text{th} \) 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 and Thomas (2006); Kapur and Kesavan (1992); Shannon (1948)) of the transition probabilities:

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

(8)

In matrix form, Equation (8) becomes

$$h = - \text{diag} \left( Q (\log Q)^T \right)$$

(9)

where the logarithm is taken elementwise. $h$ is a $(n-1) \times 1$ 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 $u_k$:

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

(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 for instance Shannon (1948); Cover and Thomas (2006)). The main difference here is that we are dealing with absorbing Markov chains which are therefore not ergodic.

The next step will be to set the global entropy $H$ to a predefined value, say $H = H_0$, therefore fixing the global randomness occuring 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 will seek the set of transition probabilities, $\pi \equiv \{p_{kk'}\}$, for which the expected cost $v_\pi(1)$ from initial state 1 to destination state $n$ is minimal while fixing the global entropy in the network,
Therefore, a convenient choice would be to weigh the entropy related to each node, \(h_k\), to a constant value. It can be formulated as a constrained optimization problem involving a Lagrange function: minimize \(v_n(1)\) subject to the constraint \(H = H_0\) (\(H\) being given by Equation (10)). In other words, we seek \(\mathbf{P}^* = \arg\min \{v_n(1)\}\) subject to the constraint \(\sum_{k=1}^{n-1} u_k h_k = H_0\).

In Appendix B, we derive the form of the optimal transition probabilities distribution within state \(k\), which appears to be a multinomial logit, or Boltzmann distribution:

\[
p_{kk'} = \frac{\exp \left[ -\frac{n_k}{\eta} (c_{kk'} + v(k')) + \frac{1}{u_k} \sum_{l \neq n} (\partial u_l / \partial p_{kk'}) h_l \right]}{\sum_{l \in S(k)} \exp \left[ -\frac{n_k}{\eta} (c_{kl} + v(l')) + \frac{1}{u_k} \sum_{l \neq n} (\partial u_l / \partial p_{kk'}) h_l \right]}, \text{ for } k \neq n \tag{11}
\]

where \(n_k\) is the \(k\)th element of \(n\) (containing as entries the first row of the fundamental matrix \(N\) – see Equation (5)) and \(v(k)\) is the minimum expected cost computed thanks to Equation (7). The parameter \(\eta > 0\) controls the entropy: the larger \(\eta\), the larger the entropy. Notice that \(\eta\) 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 \(\eta\) and \(H\) is monotonically decreasing, all other parameters being fixed), but this is not needed since \(\eta\) 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 \(\eta\) instead of \(H_0\).

### 3.1 Fixing the expected entropy spread in the network

By looking to Equation (11), we observe that, when assuming a constant weight \(u_k = 1\), 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 \(u_k = n_k\),

\[
H = \sum_{k=1}^{n-1} n_k h_k = -\sum_{k=1}^{n-1} n_k \sum_{k' \in S(k)} p_{kk'} \log p_{kk'} \tag{12}
\]

Interestingly enough, this particular choice will be shown to be similar to Akamatsu’s model that 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 \(\sum_{l \neq n} (\partial u_l / \partial p_{kk'}) h_l = \sum_{l \neq n} (\partial u_l / \partial p_{kk'}) h_l\) in Equation (11), which is done in Appendix C:

\[
\sum_{l \neq n} (\partial u_l / \partial p_{kk'}) h_l = n_k \bar{e}_k^T \mathbf{N} \mathbf{h} = n_k \sum_{l \neq n} n_{k'l} h_l \tag{13}
\]

where \(n_{k'l}\) is element \((k',l)\) of matrix \(\mathbf{N}\) and \(\bar{e}_k\) is a \((n-1) \times 1\) column vector full of 0’s except its 4th entry containing a 1. Let us rewrite Equation (11) by
setting $u_k = n_k$, and using Equation (13),

$$p_{kk'} = \frac{\exp \left[ -\frac{1}{\eta} (c_{kk'} + v(k')) + \gamma_{k'} \right]}{\sum_{l' \in S(k)} \exp \left[ -\frac{1}{\eta} (c_{kl'} + v(l')) + \gamma_{l'} \right]}, \text{ with } k \neq n$$

(14)

where \( \gamma_{k'} = \sum_{i \neq n} n_{k'li} h_i \), for \( k' \neq n \)

\[ \gamma_n = 0, \text{ for destination state } n \]

Notice that the second term in the exponential of the numerator of Equation (14), \( \gamma_{k'} \), 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 counter-balancing the expected entropy. \( \eta \) plays the role of a temperature. When the global entropy \( H \) is small, that is, \( \eta \) 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 (14) and \( \gamma_{k'} \to 0 \), with the result that the computation of the expected cost (Equation (2)) reduces to Bellman’s equations for finding the shortest path from the initial state to the destination state in this case.

Equations (7) and (14) are thus the necessary optimality conditions. The parameter \( \eta \) is supposed to be provided a priori by the user, according to the desired degree of randomness he 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,

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

(15)

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

$$p_{kk'} = \frac{\exp \left[ -\frac{1}{\eta} (c_{kk'} + v(k')) + \gamma_{k'} \right]}{\sum_{l' \in S(k)} \exp \left[ -\frac{1}{\eta} (c_{kl'} + v(l')) + \gamma_{l'} \right]}, \text{ with } k \neq n$$

(16)

where \( \gamma_{k'} = \sum_{i \neq n} n_{k'li} h_i - \left( \sum_{i \neq n} n_{k'l} \right) \left( \sum_{l' \neq n} \pi_{l'} h_{l'} \right) \), for \( k' \neq n \)

\( \gamma_n = 0, \text{ for destination state } n \)

Other choices are, of course, possible, depending on the problem at hand.
3.3 Using a reference a priori policy by fixing the Kullback 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 Kullback divergence $J$ (see, e.g., Kapur and Kesavan (1992)) between this reference policy and the policy we are seeking. Thus, we define $j_k$ as

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

and $J$ as

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

By proceeding as in previous section, we easily obtain

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

for $k \neq n$.

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, $n_k^{\text{ref}}$, that is, to set $u_k = n_k^{\text{ref}}$. This leads to the following necessary conditions

$$p_{kk'} = \frac{p_{kk'}^{\text{ref}} \exp \left[ -\frac{n_k}{n_k^{\text{ref}}} (c_{kk'} + v(k')) \right]}{\sum_{l' \in S(k)} p_{kl'}^{\text{ref}} \exp \left[ -\frac{n_k}{n_k^{\text{ref}}} (c_{kl'} + v(l')) \right]}$$

for $k \neq n$.

4 Computation of the optimal policy

Equations (7) and (14) suggest a simple iterative procedure, similar to the well-known value-iteration algorithm, for the computation of the policy. This algorithm is detailed in Algorithm 1 and will be used in all our experiments. 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
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 controlling the degree of randomness.
- \( C = \begin{bmatrix} D & s \\ \infty^T & 0 \end{bmatrix} \): the \( n \times n \) cost matrix; node \( n \) is the destination node.

1. Initialize \( P \leftarrow \begin{bmatrix} Q & r \\ 0^T & 1 \end{bmatrix} \), 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 \( (I - Q) \) is not of full rank then
      4. return Error: the fundamental matrix is not invertible.
      5. end if
   6. \( h \leftarrow -\text{diag}(Q[\log Q]^T) \) \{see Equation (9)\}.
   7. Solve \( (I - Q) \tilde{v} = h \) with respect to \( \tilde{v} \) \{see Equation (14)\}.
   8. \( \tilde{v} \leftarrow \begin{bmatrix} \tilde{v} \\ 0 \end{bmatrix} \)
   9. Solve \( (I - Q) v = \text{diag}(QD^T + rs^T) \) with respect to \( v \) \{see Equation (7)\}.
   10. \( v \leftarrow \begin{bmatrix} v \\ 0 \end{bmatrix} \)

11. \[
    p_{kk'} = \exp \left[ -\frac{1}{\eta} (c_{kk'} + v(k')) + \kappa_{k'} \right] 
      \left( \sum_{l' \in S(k)} \exp \left[ -\frac{1}{\eta} (c_{k'l'} + v(l')) + \kappa_{l'} \right] \right), \text{ for all } k \neq n, k' 
    p_{nk'} = \delta_{nk'}, \text{ for destination node } n 
    \]
    \{see Equation (14)\}.

12. until convergence of \( P \).
13. return the policy \( P \) of the form \( \begin{bmatrix} Q & r \\ 0 & 1 \end{bmatrix} \): the transition-probabilities matrix containing the elements \( p_{kk'} \).

following Lagrange function:

\[
L = v(1) + \sum_{k \neq n} \lambda_k \left[ v(k) - \sum_{k' \in S(k)} p_{kk'} (c_{kk'} + v(k')) \right] 
+ \lambda_n [v(n) - 0] + \sum_{k \neq n} \mu_k \left[ \sum_{k' \in S(k)} p_{kk'} - 1 \right] 
+ \eta \left[ \sum_{k \neq n} n_k \sum_{k' \in S(k)} p_{kk'} \log p_{kk'} + H_0 \right] 
+ \sum_{k' \neq n} \phi_{k'} \left[ n_{k'} - \sum_{k \neq n} p_{kk'} n_k - \delta_{kk'} \right], \quad (21)
\]
which is identical to the Lagrange function defined in Appendix B, Equation (53), except that the formula for computing the expected number of passages, \( n_k \), is now made explicit, so that the \( n_k \) can now be considered as independent variables. Indeed, \( n = N^T e_1 \), from which we deduce \( n = Q^T n + e_1 \) and thus \( n_k' = \sum_{k \neq n} n_{k'k} n_k - \delta_{k'k} \). Notice also that the \( \eta \) 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 Appendix B and displayed in Equation (14).

Setting the partial derivative of \( \mathcal{L} \) in terms of \( \mu_k \) and \( p_{kk'} \) equal to zero while fixing all the other parameters provides the update equations for the transition probabilities matrix \( P \), which is uniquely attained. On the other hand, setting the partial derivative in terms of the other parameters (except \( \eta \), considered as fixed) equal to zero provides the re-estimation equations for the expected costs \( v \) and the expected entropies \( \kappa \) (we easily find \( \phi_k = \eta \kappa_k \)), which are both uniquely attained provided the matrix \( (I - Q) \) 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), p. 268), which is indeed the case. Moreover, it is shown 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 Subsection 6.3 an informal proof showing that the stationary point obtained by the Algorithm 1 must be a global minimum.

Computationally speaking, it is very demanding since it requires, in addition to be iterative, either the computation of the fundamental matrix, \( N = (I - Q)^{-1} \), or the solution of two linear systems of equations at each iteration. Section 5 will introduce 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 of 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 and 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 for instance Jaynes (1957); Reichl (1998); Schrodinger (1952)), allowing 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 \( \varphi_r \) from the initial state to the destination state has a probability of being followed proportional to \( \exp[-\theta E(\varphi_r)] \), where \( E(\varphi_r) = E_r \) is the total cost associated to the \( r \)th path, and referred to as the energy associated to that path. In other words, short paths are favoured while long paths are penalized. The parameter \( \theta \) 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 (1)), we consider that the total cost associated to a path is additive, i.e.

\[
E(\varphi_r) = \sum_{t=0}^{t_f} c_{s_t s_{t+1}} (22)
\]

where \( s_0 = k_0 \) is the initial state and \( s_{t_f} = d \) is the destination state. Here, we assume that \( \varphi_r \) 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 \( d \). Remember also that the destination state \( d \) is made absorbing so that an infinite cost is associated to 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 \( \varphi_r \) is thus

\[
P(\varphi_r) = \frac{\exp[-\theta E(\varphi_r)]}{\sum_{r=1}^{\infty} \exp[-\theta E(\varphi_r)]} (23)
\]

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 (23) is defined, will be denoted by \( R \).

Let us now show that all the quantities of interest can be computed from a quantity, \( Z = \sum_{r=1}^{\infty} \exp[-\theta E(\varphi_r)] \), the denominator of Equation (23), which corresponds to the partition function in statistical physics (see for instance Jaynes (1957); Reichl (1998); Schrödinger (1952)) or Markov random fields (Rue and Held (2005)). The main point is that the partition function can easily be computed from the immediate cost matrix \( C \) (see the next section). We further define the free energy \( F \) as \( F = -\frac{1}{\theta} \log(Z) = -T \log(Z) \) 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:

\[
\mathbb{E} = \frac{\theta (-\log(Z))}{\theta} = \frac{\sum_{r=1}^{\infty} \exp[-\theta E(\varphi_r)] E(\varphi_r)}{\sum_{r=1}^{\infty} \exp[-\theta E(\varphi_r)]} = \sum_{r=1}^{\infty} P(\varphi_r) E(\varphi_r) (24)
\]

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

\[
\mathbb{E} \left\{ (E - \mathbb{E})^m \right\} = (-1)^m \frac{\partial^m}{\partial \theta^m} (\log(Z)) (25)
\]
where $E \{ \cdot \}$ is the expectation operator.

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

$$
\eta_{kk'} = \frac{\partial F}{\partial c_{kk'}} = \frac{1}{\theta} \frac{\partial (-\log(Z))}{\partial c_{kk'}} = \sum_{r=1}^{\infty} \frac{\exp[-\theta E(\varphi_r)]}{Z} \delta(r; k, k') = \sum_{r=1}^{\infty} P(\varphi_r) \delta(r; k, k')
$$

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

$$
p_{kk'} = \frac{\sum_{r=1}^{\infty} P(\varphi_r) \delta(r; k, k')}{\sum_{l=1}^{\infty} \sum_{r=1}^{\infty} P(\varphi_r) \delta(r; k, l) / \sum_{l=1}^{\infty} \eta_{kl}}
$$

(27)

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

$$
\eta_k = \sum_{l=1}^{n} \eta_{lk}
$$

(28)

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 $\varphi_r \in \mathcal{R}$ as (Ekroot and Cover (1993)):

$$
H = - \sum_{r=1}^{\infty} P(\varphi_r) \log P(\varphi_r)
$$

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

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

$$
= \log Z + \frac{1}{T} \sum_{r=1}^{\infty} E(\varphi_r) \frac{\exp[-E(\varphi_r)/T]}{Z} = \log Z + \frac{E}{T}
$$

(29)

The entropy $H$ can also be found by

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

$$
= \log Z + \sum_{r=1}^{\infty} \frac{E(\varphi_r)}{T} \frac{\exp[-E(\varphi_r)/T]}{Z} = \log Z + \frac{E}{T}
$$

(30)

which is indeed the same as Equation (29).
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, $C$, from which we build a new matrix, $W$, which simply contains the exponentials of the costs $c_{ij}$, that is, of the elements of $C$,

$$W = \exp[-\theta C], \quad (31)$$

where the exponential is taken elementwise. Now, we easily observe that element $(k_0, d)$ of the matrix $W^t$ (where $W$ is raised to the power $t$) is

$$[W^t]_{k_0d} = \sum_{r \in \mathcal{R}(t)} \exp[-\theta E(\mathcal{P}_r)]$$

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

$$\mathcal{Z} = \sum_{t=1}^{\infty} \sum_{r \in \mathcal{R}(t)} \exp[-\theta E(\mathcal{P}_r)] = \left[ \sum_{t=1}^{\infty} W^t \right]_{k_0d} \quad (32)$$

which converges if the spectral radius of $W$, $\rho(W)$, is less than 1 and depends on the parameter $\theta$. We will therefore assume in the sequel that $\rho(W) < 1$. Since the matrix $W$ only contains non-negative elements, a sufficient condition for $\rho(W) < 1$ is that all its row sums are less than 1, which can always be achieved for a sufficiently large value of $\theta$ (see Equation (31)) and thus a sufficiently low entropy. Indeed, it is well-known that the spectral radius of a real square matrix is always smaller or equal than its maximum absolute row sum norm (see, e.g., Bronson (1989), p. 111).

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

$$W = \begin{bmatrix} Q_{wr} & r_w \\ 0^T & 0 \end{bmatrix} \quad (33)$$

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 ($n$th) absorbing state and therefore put a 0 on every entry of this last row. Computing the series of powers of $W$ provides

$$\sum_{t=1}^{\infty} W^t = (I - W)^{-1} - I \quad (34)$$

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

$$\mathcal{Z} = [(I - W)^{-1} - I]_{1n} = e_1^T [(I - W)^{-1} - I] e_n$$

$$= e_1^T (I - W)^{-1} e_n = [Z]_{1n} = z_{1n} \quad (35)$$

where we posed $Z = (I - W)^{-1} = I + W + W^2 + \cdots$. Notice that since the last row of $W$ is $0^T$, so are the powers of $W$, so that $z_{nn} = 1$. Matrix $Z$
will be called the fundamental matrix by analogy with the theory of absorbing Markov chains (see the definition of matrix $N$ after Equation (5)). Notice that if the cost matrix $C$ is symmetric (undirected network), since it is assumed that $\rho(W) < 1$, the matrix $(I - W)$ is positive definite.

5.3 Computation of the main quantities

We now have to compute the derivatives of $Z$ (assuming the paths probability distribution (23)) in terms of $\theta$ and $c_{kk'}$ in order to obtain the different quantities of interest (Equations (24)-(29)), which is done in Appendix D. For the expected energy or cost, we obtain

$$E = \frac{\partial \left( -\log Z \right)}{\partial \theta} = -\frac{e^TZW'_\theta Ze_n}{z_{1n}}$$

(36)

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

$$\eta_{kk'} = \frac{\partial F}{\partial c_{kk'}} = \frac{z_{1k}z_{kn} \exp[-\theta c_{kk'}]}{z_{1n}}, \text{ with } k \neq n$$

(37)

The expected number of passages through state $k$ is

$$\pi_k = \frac{z_{1k}z_{kn}}{z_{1n}}, \text{ with } k \neq n$$

(38)

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 (Equation (27)) are provided by

$$p_{kk'} = \frac{z_{k'k} \exp[-\theta c_{kk'}]}{z_{kn}}, \text{ with } k \neq n$$

(39)

And finally, the entropy spread in the network is

$$H = -\frac{\partial F}{\partial T} = \log Z + \frac{E}{T} = \log z_{1n} + \theta E$$

(40)

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 (i) the two objective functions (see Section 6.1 below) and (ii) the two constraints on the entropy spread in the network (see Section 6.2 below) refer to the same quantities. Thus, both models minimize the same objective function, subject to the same entropy equality constraint, either with respect to the transition probabilities (first model; Section 3), or with respect to the paths probabilities (second model; see Section 6.3 below). 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 (Equation (23)).
6.1 Equivalence of the objective function

As objective function, the first model minimizes the expected cost expressed as

\[ v_\pi(1) = E_\pi \{ \sum_{t=0}^{\infty} c_{s_t,s_{t+1}} | s_0 = k_0 \} \]  
(Equation (1)), the value of this expected cost being provided by the recurrence relation of Equation (2). It is shown 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

\[ E(\varphi_r) = \sum_{t=0}^{t_f} c_{s_t,s_{t+1}} \]  
(Equation 22), the expected cost \( v_\pi(1) \) can be re-expressed in terms of the paths probability distribution:

\[
v_\pi(1) = E_\pi \left\{ \sum_{t=0}^{\infty} c_{s_t,s_{t+1}} | s_0 = k_0 \right\} = E_\pi \left\{ \sum_{t=0}^{t_f} c_{s_t,s_{t+1}} | s_0 = k_0 \right\} = \sum_{r=1}^{\infty} \Pr(\varphi_r) E(\varphi_r), \varphi_r \in \mathcal{R} \tag{41}
\]

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

\[
\Pr(\varphi_r) = \prod_{k,k'} (p_{kk'})^{\delta(r;k,k')}
\tag{42}
\]

is the probability of a trajectory \( \varphi_r \in \mathcal{R} \) 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 \( r, \varphi_r \). We thus end up in (41) with the expected energy, as defined in Equation (24), provided \( \Pr(\varphi_r) = P(\varphi_r) \) (where \( P(\varphi_r) \) is given by Equation (23)), 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 (41).

Let us now prove that the probability of a path is equivalent in both formulations, that is, \( \Pr(\varphi_r) = P(\varphi_r) \) when Equation (39) 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

\[
\Pr(\varphi_r) = \prod_{k,k'} (p_{kk'})^{\delta(r;k,k')}
\]

\[
= \frac{z_{kn}}{z_{kn}} \exp \left[ -\theta c_{kk'} \right]^{\delta(r;k,k')}
\]

\[
= \frac{z_{kn}}{z_{kn}} \prod_{k,k'} [\exp \left[ -\theta c_{kk'} \right]]^{\delta(r;k,k')}
\]

\[
= \frac{z_{nn}}{z_{nn}} \exp \left[ -\theta \sum_{k,k'} c_{kk'} \delta(r;k,k') \right]
\]

\[
= \frac{1}{Z} \exp \left[ -\theta E(\varphi_r) \right] = P(\varphi_r) \tag{43}
\]
where we used Equation (39) as well as $E(\wp_r) = \sum_{k,k'} c_k k' \delta(r; k, k')$, $z_{nn} = 1$ and $z_1 = Z$. This shows that the paths probability distribution (23) 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 $R$, but not necessarily of the form (23). 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 (12) and (29), express the same quantity. In a second paper, Akamatsu (1997) proved that the entropy defined by Equation (29) 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 and Thomas (2006)), in order to show the equivalence between the entropy concepts. We easily find

$$H = -\sum_{r=1}^{\infty} P(\wp_r) \log P(\wp_r)$$

$$= -\sum_{r=1}^{\infty} P(\wp_r) \log \left[ \prod_{k,k'} (p_{kk'})^{\delta(r; k, k')} \right]$$

$$= -\sum_{r=1}^{\infty} P(\wp_r) \sum_{k,k'}^{n} \delta(r; k, k') \log p_{kk'}$$

$$= -\sum_{k,k'}^{n} \left[ \sum_{r=1}^{\infty} P(\wp_r) \delta(r; k, k') \right] \log p_{kk'}$$

$$= -\sum_{k,k'}^{n} [\pi_{kk'}] \log p_{kk'} = -\sum_{k,k'}^{n} \pi_{kk'} \log p_{kk'}$$

$$= -\sum_{k=1}^{n} \pi_{kk} \sum_{k'=1}^{n} p_{kk'} \log p_{kk'}$$ (44)

Notice that we used $\sum_{k=1}^{n} \pi_{kk'} = \pi_{kk} p_{kk'}$ as well as the definitions introduced in Section 5.1. We also needed $0 \log 0 = 0$ and $0^0 = 1$. Now, Equation (44) is exactly the entropy defined in Equation (12). 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, $\{P(\wp_r)\}$ (the paths probability distribution), minimizing $E = \sum_{r=1}^{\infty} P(\wp_r) E(\wp_r)$ subject to
the constraint $- \sum_{r=1}^{\infty} P(\varphi_r) \log P(\varphi_r) = H_0$. By defining the Lagrange function

$$L = \sum_{r=1}^{\infty} P(\varphi_r) E(\varphi_r) + \lambda \left[ \sum_{r=1}^{\infty} P(\varphi_r) \log P(\varphi_r) + H_0 \right] + \mu \left[ \sum_{r=1}^{\infty} P(\varphi_r) - 1 \right], \quad (45)$$

we not surprisingly find the Boltzmann distribution

$$P(\varphi_r) = \frac{\exp \left[ - \theta E(\varphi_r) \right]}{\sum_{r=1}^{\infty} \exp \left[ - \theta E(\varphi_r) \right]} = \frac{1}{Z} \exp \left[ - \theta E(\varphi_r) \right] \quad (46)$$

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, (46) corresponds exactly to the paths probability distribution that was assumed in Akamatsu’s model (Equation (23)). Notice that if the constraint involves Kullback-Leibler’s divergence with respect to a reference policy, as in Section 3.3, instead the 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 (1), subject to an entropy constraint (10). For the second (sum-over-paths) model, the optimal paths probabilities $P(\varphi_r)$ are found instead by minimizing the objective function (22) subject to the entropy constraint $H_0 = - \sum_{r=1}^{\infty} P(\varphi_r) \log P(\varphi_r)$. The solution for this second formulation is unique and corresponds to the Boltzmann distribution (Equation (46)); 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 (37). This arc flow is also unique provided that $\rho(W) < 1$. In the case of a near-zero entropy, if there exists several shortest paths between the initial and the destination node, according to (46), 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 Equation (39)). 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 (Equation (14)) plays exactly the same role as the parameter $\theta$ in the sum-over-paths model (Equation (23)).

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 (42)) and vice-versa (see Equation (27)), 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 $\rho(W) < 1$. First, denote as $P_1$ 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 (42)). 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 (43)) that the global minimum of \( E \) in terms of the \( \mathcal{P}(\varphi_r) \) belonging to the complete set \( \mathcal{P} \), and provided by Equation (46), 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 (46)). Furthermore, we already know that \( \pi^* \) is a stationary point of \( v_\pi(1) \) (Section 4). The corresponding paths probabilities, \( \{\mathcal{P}^*(\varphi_r)\} \), obtained from \( \pi^* \) by (42), must also be a stationary point of \( E \) 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, \( \{\mathcal{P}^*(\varphi_r)\} \) must be a global minimum, and so does \( \pi^* \). 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.

Notice 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 \((I - W)\) can be a serious issue. However, by examining the Equations (36)-(40), we immediately notice that only \( Z^T e_1 = z_1 \) and \( Z e_n = z_n \) need to be computed. These two quantities can be found by solving the linear systems of equations

\[
(I - W)^T z_1 = e_1 \quad \text{and} \quad (I - W)z_n = e_n \tag{47}
\]

Notice that these equations could also be solved iteratively by the updating equations \( z_1 \leftarrow W^T z_1 + e_1 \) and \( z_n \leftarrow W z_n + e_n \) since we assume \( \rho(W) < 1 \).

In other words, the column vector \( z_1 = (\text{row}_1(Z))^\top \) contains the elements of the first row of matrix \( Z \) while the column vector \( z_n = \text{col}_1(Z) \) contains the elements of the last column of \( Z \). These linear systems of equations can be solved efficiently, especially when the matrix \( W \) is sparse (Davis (2006)), which is often the case. Remember also that for undirected graphs, the matrix \( W \) is positive definite.

Elementwise, these last equations yield

\[
\begin{cases}
z_{11} = 1 \\
z_{1k'} = \sum_{k \in \mathcal{P}(k')} \exp[-\theta c_{kk'}] z_{1k}
\end{cases}
\tag{48}
\]
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 randomness.
- \( C = \begin{bmatrix} D & s \\ \infty & \infty \end{bmatrix} \): the \( n \times n \) cost matrix.

1. \( W = \exp \left[ -\theta C \right] \) \{Elementwise exponential; see Equation (31)\}
2. if \( \rho(W) \geq 1 \) then
   3. return Error: the spectral radius is greater than one.
4. end if
5. Solve \((I - W)z_n = e_n\) with respect to \( z_n \). The elements of \( z_n \) are \( z_{kn}. \) \{see Equation (47)\}
6. Compute \( p_{kk'} = \frac{z_{k'n}}{z_{kn}} \exp \left[ -\theta c_{kk'} \right], \) for all \( k \neq n, k' \) \{see Equation (39)\}
7. return the policy \( P \) of the form \( \begin{bmatrix} Q & r \\ 0^T & 1 \end{bmatrix} \): the transition-probabilities matrix containing the elements \( p_{kk'} \).

and

\[
\begin{cases} 
  z_{nn} = 1 \\
  z_{kn} = \sum_{k' \in S(k)} \exp \left[ -\theta c_{kk'} \right] z_{k'n}
\end{cases}
\]

(49)

where, as before, \( P(k') \) is the set of predecessors of node \( k' \) and \( S(k) \) is the set of successors of node \( k \).

There is an interesting similarity between the Equations (37) & (48)-(49) and the forward/backward procedure for computing the forward/backward variables when estimating transition probabilities of a hidden Markov model (Jelinek (1997); Rabiner and Juang (1993)). These equations are also similar to the estimation equation obtained for conditional random fields (Lafferty et al. (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 and Stutzle (2004)) where the amount of pheromone dropped at each node \( k \) is represented by \( z_{kn}. \)

Consequently, we obtain the algorithm displayed in Algorithm 2, allowing to compute the transition-probabilities matrix \( P \). Remember that we assume that the user supplies the parameter \( \theta \) in place of the entropy value \( H_0. \)

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 to simplify 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 \((I - W)\) is upper triangular and the matrix \((I - W)^T\) 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 for instance Sedgewick (1990)). In this special case, the linear equations defined in Equations (47) can easily be solved by simple back-substitution. For instance, the Equations (48) and (49) 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/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 4 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 1: First network used in our simulations. The immediate costs are indicated on the arcs.](image)

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

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 $H$ ($= 0.5, 1.0, 1.5, 2.0$).

![Figure 3: Resulting Markov chains together with the transition probabilities and the average cost, for four different values of the global entropy.](image)

**8.2 Second experiment**

The second experiment is performed on the network shown in Figure 4. It is composed of 8 nodes connected by edges of different weights, representing costs.

As for 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.
8.3 Third experiment

The third experiment is performed on a $40 \times 30$ 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 round 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.

The resulting expected number of passages through every cell of the grid for three values of $\theta$, provided by Equation (38), is shown in Figure 7.
Figure 6: Resulting Markov chains together with the transition probabilities and the average cost, for four different values of the global entropy.

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 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 to 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, during the time of submission of this paper,
we already 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/alignment paths. Yet another application would be to extend some of the results to the so-called semiring framework (Carre (1979); Gondran and Minoux (1984); Mohri (2002)). Finally, we also plan to investigate the links between our proposed models and the recent work of Nesterov (2007).

Acknowledgments

Part of this work has been funded by projects with the “Région wallonne” and the “Région de Bruxelles-Capitale”. We thank Masashi Shimbo for pointing out the relationships to conditional random fields and Francois Glineur for his useful insights on optimization theory.

References


APPENDIX: PROOF OF THE MAIN RESULTS

A Computation of the expected cost-to-go in terms of the fundamental matrix

Let us start from Equation (3) which is restated here,

\[ v = \text{diag}(PC^T) + Pv \]

Thus, we have

\[ \begin{bmatrix} \tilde{v} \\ 0 \end{bmatrix} = \text{diag} \left( \begin{bmatrix} Q & r \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} D^T & \infty \\ s^T & 0 \end{bmatrix} \right) + \begin{bmatrix} Q & r \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ 0 \end{bmatrix}, \]

(50)

from which we easily deduce

\[ \tilde{v} = \text{diag}(QD^T + rs^T) + Q\tilde{v} \]

(51)

Isolating \( \tilde{v} \) provides us with the required result:

\[ \tilde{v} = (I - Q)^{-1} \text{diag}(QD^T + rs^T) \]

\[ = N \text{diag}(QD^T + rs^T) \]

(52)

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

B 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 (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 (Equation (2); parameters \( \lambda_k \)), the sum-to-zero constraints for the transition probabilities at each node (parameters \( \mu_k \)) and the entropy constraint (Equation (10); parameter \( \eta \)),

\[ \mathcal{L} = v(1) + \sum_{k \neq n} \lambda_k \left[ v(k) - \sum_{k' \in S(k)} p_{kk'} (c_{kk'} + v(k')) \right] \]

\[ + \lambda_n [v(n) - 0] + \sum_{k \neq n} \mu_k \left[ \sum_{k' \in S(k)} p_{kk'} - 1 \right] \]

\[ + \eta \left[ \sum_{k \neq n} u_k \sum_{k' \in S(k)} p_{kk'} \log p_{kk'} + H_0 \right] \]

(53)

Strictly speaking, a constraint guaranteeing the positivity of the transition probabilities should be introduced as well, but this is actually not necessary since the resulting transition probabilities will automatically be positive.
Differentiating this Lagrange function in terms of the transition probabilities, $\partial \mathcal{L}/\partial p_{l'}$, and equating to zero gives

\[-\lambda_l(c_{l'} + v(l')) + \mu_l + \eta u_l (\log p_{l'} + 1) - \eta \sum_{k \neq n} (\partial u_k / \partial p_{l'}) h_k = 0 \tag{54}\]

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

\[\eta u_l \log p_{l'} = -\mu_l - \eta u_l + \lambda_l (c_{l'} + v(l')) + \eta \sum_{k \neq n} (\partial u_k / \partial p_{l'}) h_k \tag{55}\]

Exponentiating this last equation yields

\[p_{l'} = \exp \left[ -\frac{\mu_l - \eta u_l}{\eta u_l} \right] \exp \left[ \frac{\lambda_l (c_{l'} + v(l')) + \eta \sum_{k \neq n} (\partial u_k / \partial p_{l'}) h_k}{\eta u_l} \right] \tag{56}\]

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:

\[\exp \left[ -\frac{\mu_l - \eta u_l}{\eta u_l} \right] = \left[ \sum_{l' \in S(l)} \exp \left[ \frac{\lambda_l (c_{l'} + v(l')) + \eta \sum_{k \neq n} (\partial u_k / \partial p_{l'}) h_k}{\eta u_l} \right] \right]^{-1} \tag{57}\]

By replacing (57) in Equation (56) and defining $\theta_k = -\lambda_k$, we finally obtain

\[p_{kk'} = \frac{\exp \left[ -\frac{\theta_k}{\eta u_k} (c_{kk'} + v(k')) + \frac{1}{u_k} \sum_{l' \neq n} (\partial u_l / \partial p_{kk'}) h_l \right]}{\sum_{l' \in S(k)} \exp \left[ -\frac{\theta_k}{\eta u_k} (c_{kk'} + v(k')) + \frac{1}{u_k} \sum_{l' \neq n} (\partial u_l / \partial p_{kk'}) h_l \right]} \tag{58}\]

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

\[\lambda_l = \sum_{k \in P(l)} p_{kl} \lambda_k, \text{ for } l \neq 1, \tag{59}\]

$P(l)$ 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)

\[\lambda_1 = -1 \tag{60}\]

Defining $\theta_k = -\lambda_k$ provides

\[\theta_l = \sum_{k \in P(l)} p_{kl} \theta_k, \text{ for } l \neq 1 \tag{61}\]

\[\theta_1 = 1, \text{ for initial state} \]

35
Rewriting this last Equation (61) in matrix form gives \( \theta = P^T \theta + e_1 \). Therefore, when considering only the transient states, \( \tilde{\theta} = Q^T \tilde{\theta} + \tilde{e}_1 \). In other words, we have
\[
\tilde{\theta} = (I - Q^T)^{-1} \tilde{e}_1 \tag{62}
\]

By comparing Equation (62) with Equation (5), we easily observe that the Lagrange parameters vector \( \tilde{\theta} \) simply corresponds to the expected number of visits to each transient state before reaching the absorbing state. In other words, \( \tilde{\theta} = n \) in Equation (58).

Finally, expressing the fact that the policy has a fixed entropy,
\[
- \sum_{k \neq n} n_k \log p_{kk'} = H_0, \tag{63}
\]
allows us to compute the value of \( \eta \) in terms of \( H_0 \).

## C 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
\[
\sum_{l \neq n} (\partial n_l / \partial p_{kk'}) h_l = \frac{\partial n^T}{\partial p_{kk'}} h
\]
\[
= \frac{\partial (N^T \tilde{e}_1)^T}{\partial p_{kk'}} h
\]
\[
= \tilde{e}_1^T \frac{\partial N}{\partial p_{kk'}} h \tag{64}
\]
where we used \( n = N^T \tilde{e}_1 \) (see Equation (5)).

But since \( N = (I - Q)^{-1} \),
\[
\frac{\partial N}{\partial p_{kk'}} = \frac{\partial (I - Q)^{-1}}{\partial p_{kk'}}
\]
\[
= -(I - Q)^{-1} \frac{\partial (I - Q)}{\partial p_{kk'}} (I - Q)^{-1}
\]
\[
= N \frac{\partial Q}{\partial p_{kk'}} N
\]
\[
= N \tilde{e}_k \tilde{e}_k^T N \tag{65}
\]
where we used the standard formula for matrix differentiation, \( dM^{-1} = -M^{-1}(dM)M^{-1} \) (see, e.g., Harville (1997)). Replacing (65) in (64) provides
\[
\sum_{l \neq n} (\partial n_l / \partial p_{kk'}) h_l = \tilde{e}_1^T N \tilde{e}_k \tilde{e}_k^T N h
\]
\[
= n^T \tilde{e}_k \tilde{e}_k^T N h
\]
\[
= n_k \tilde{e}_k^T N h
\]
\[
= n_k \sum_{l \neq n} n_{k'1} h_l \tag{66}
\]

36
which is the required result.

Now, when \( k' = n \) (absorbing node), since the \( n_t \) do not depend explicitly on the \( p_{kn} \) \((n_t \) only depends on \( Q \)), \( \partial n_t / \partial p_{kn} = 0 \) and thus \( \sum_{t \neq n} (\partial n_t / \partial p_{kn}) h_t = 0 \).

**D Computation of the derivatives of the partition function \( Z \)**

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 \( Z = \left( (I - W)^{-1} \right)_{1n} \).

First, we have to compute \( E = \frac{\partial(-\log Z)}{\partial \theta} \)

\[
E = \frac{\partial(-\log Z)}{\partial \theta} = -\frac{\partial}{\partial \theta} \left[ e_1^T (I - W)^{-1} e_n \right] = -\frac{e_1^T \partial (I - W)^{-1} e_n}{e_1^T (I - W)^{-1} e_n} \quad (67)
\]

Let us compute \( \partial (I - W)^{-1} \); by setting \( Z = (I - W)^{-1} \) and denoting element \( i, j \) of \( Z \) by \( z_{ij} \), we obtain

\[
\partial (I - W)^{-1} = -Z \partial (I - W) Z \\
= Z \partial (I - W) Z \\
= ZW_\theta' Z \quad (68)
\]

where the matrix \( W_\theta' = \partial W / \partial \theta \) contains the elements \( [W_\theta'_{ij}] = -c_{ij} \exp \left[ -\theta c_{ij} \right] \).

Therefore, the expected cost needed for reaching state \( n \) from state \( 1 \), \( E \), is

\[
E = -\frac{e_1^T Z W_\theta' Z e_n}{e_1^T Z e_n} = -\frac{e_1^T Z W_\theta' Z e_n}{z_{1n}} \quad (69)
\]

We now turn to the computation of \( \eta_{kk'} \). We easily find

\[
\eta_{kk'} = \frac{\partial E}{\partial c_{kk'}} = \frac{1}{\theta} \frac{\partial \log(Z)}{\partial c_{kk'}} \\
= \frac{1}{\theta} \frac{\partial}{\partial c_{kk'}} \left[ e_1^T (I - W)^{-1} e_n \right] \\
= \frac{1}{\theta} e_1^T \partial_{c_{kk'}} (I - W)^{-1} e_n \\
= \frac{1}{\theta} e_1^T \left( e_1^T (I - W)^{-1} e_n \right) e_1 \\
= \frac{1}{\theta} e_1^T \partial_{c_{kk'}} Z e_n = \frac{1}{\theta} e_1^T (\partial_{c_{kk'}} Z) e_n \quad (70)
\]

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

\[
\partial_{c_{kk'}} (I - W)^{-1} = -Z \partial_{c_{kk'}} (I - W) Z \\
= Z \partial_{c_{kk'}} W Z \\
= -\theta \exp \left[ -\theta c_{kk'} \right] Z e_1 e_1^T Z \\
= -\theta Z e_1 e_1^T Z \quad (71)
\]

Thus, \( \eta_{kk'} \) is given by

\[
\eta_{kk'} = \frac{e_1^T Z e_1^T Z e_n}{z_{1n}} = \frac{z_{1k} z_{kn} \exp \left[ -\theta c_{kk'} \right]}{z_{1n}}
\]
The expected number of passages through state $k'$ is given by

$$\pi_{k'} = \frac{z_{k'n}}{z_{1n}} \sum_{k=1}^n z_{1k} \exp \left[ -\theta_{ckk'} \right] = \frac{z_{1k'} z_{k'n}}{z_{1n}}$$  \hspace{1cm} (72)

where we used $z_{1k'} = \sum_{k=1}^n z_{1k} \exp \left[ -\theta_{ckk'} \right]$, which directly follows from Equation (48).

The transition probabilities are

$$p_{kk'} = \frac{\pi_{kk'}}{\pi_k} = \frac{z_{k'n}}{z_{kn}} \exp \left[ -\theta_{ckk'} \right]$$  \hspace{1cm} (73)

which corresponds exactly to the result of Akamatsu, derived in completely different way (see Akamatsu (1996), p. 375).

Finally, the formula for the entropy has already been derived at the end of Section 5.1; see Equations (29)-(30).