The Stochastic Shortest Path Problem: A polyhedral combinatorics perspective

Matthieu Guillot, Gautier Stauffer

The Stochastic Shortest Path Problem: A polyhedral combinatorics perspective

Matthieu Guillot and Gautier Stauffer

Univ. Grenoble Alpes, G-SCOP, 38000 Grenoble, France.
{matthieu.guillot,gautier.stauffer}@g-scop.grenoble-inp.fr

Abstract. In this paper, we give a new framework for the stochastic shortest path problem in finite state and action spaces. Our framework generalizes both the frameworks proposed by Bertsekas and Tsitsiklis [7] and by Bertsekas and Yu [8]. We prove that the problem is well-defined and (weakly) polynomial when (i) there is a way to reach the target state from any initial state and (ii) there is no transition cycle of negative costs (a generalization of negative cost cycles). These assumptions generalize the standard assumptions for the deterministic shortest path problem and our framework encapsulates the latter problem (in contrast with prior works). In this new setting, we can show that (a) one can restrict to deterministic and stationary policies, (b) the problem is still (weakly) polynomial through linear programming, (c) Value Iteration and Policy Iteration converge, and (d) we can extend Dijkstra's algorithm.

1 Introduction

The Stochastic Shortest Path problem (SSP) is a Markov Decision Process (MDP) that generalizes the classic deterministic shortest path problem. We want to control an agent, who evolves dynamically in a system composed of different *states*, so as to converge to a predefined *target*. The agent is controlled by taking *actions* in each time period[1] : actions are associated with costs and transitions in the system are governed by probability distributions that depend exclusively on the previous action taken and are thus independent of the past. We focus on finite state/action spaces : the goal is to choose an action for each state, a.k.a. a *deterministic and stationary policy*, so as to minimize the total expected cost incurred by the agent before reaching the (absorbing) target state, when starting from a given initial state.

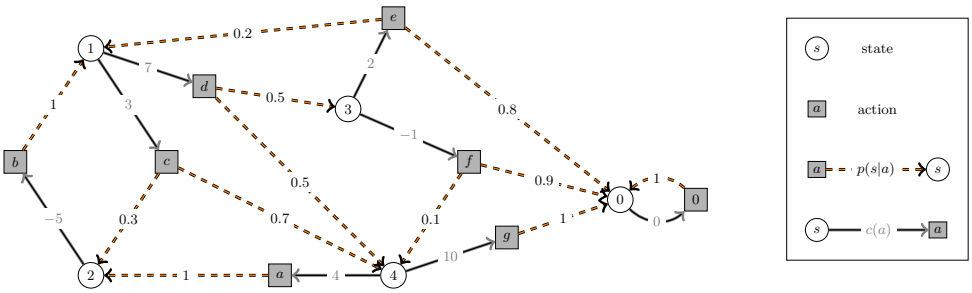

Fig. 1. A graphical representation of a SSP (with target state 0) : circles are states, squares are actions, dashed arrows indicate state transitions (probabilities) for a given action, and black edges represent actions available in a given state with corresponding cost.

More formally, a stochastic shortest path instance is defined by a tuple $(\mathcal{S}, \mathcal{A}, J, P, c)$ where $\mathcal{S} = \{0, 1, \ldots, n\}$ is a finite set of *states*, $\mathcal{A} = \{0, 1, \ldots, m\}$ is a finite set of *actions*, J is a 0/1 matrix with m lines and n columns and general term $J(a, s)$, for all $a \in \{1, ..., m\}$ and $s \in \{1, ..., n\}$, with $J(a, s) = 1$ if and only if action a is available in state s, P is a *row substochastic matrix* with m lines and n columns and general term $P(a, s) := p(s|a)$ (probability of ending in s when taking action a), for all $a \in \{1, ..., m\}$, $s \in \{1, ..., n\}$, and a cost vector $c \in \mathbb{R}^m$. The state 0 is called the *target* state and the action 0 is the unique action available in that state. Action 0 lead to state 0 with probability 1. When confusion may arise, we

[1] We focus here on discrete time (infinite) horizon problems.

denote state 0 by $0_{\mathcal{S}}$ and action 0 by $0_{\mathcal{A}}$. A *row substochastic* matrix is a matrix with nonnegative entries so that every row adds up to at most 1. We denote by $\mathcal{M}_{\leq}(l,k)$ the set of all $l \times k$ row substochastic matrices and by $\mathcal{M}_{=}(l,k)$ the set of all row stochastic matrices (*i.e.* for which every row adds up to exactly 1). In the following, we denote by $\mathcal{A}(s)$ the set of actions available from $s \in \{1,...,n\}$ and we assume without loss of generality[2] that for all $a \in \mathcal{A}$, there exists a unique s such that $a \in \mathcal{A}(s)$. We denote by $\mathcal{A}^{-1}(s)$ the set of actions that lead to s i.e. $\mathcal{A}^{-1}(s) := \{a : P(a,s) > 0\}$.

We can associate a directed bipartite graph $G = (\mathcal{S}, \mathcal{A}, E)$ with $(\mathcal{S}, \mathcal{A}, J, P)$ by defining $E := \{(s,a) : s \in \mathcal{S} \setminus \{0\}, a \in \mathcal{A} \setminus \{0\}$ with $J(a,s) = 1\} \cup \{(a,s) : s \in \mathcal{S} \setminus \{0\}, a \in \mathcal{A}^{-1}(s)\} \cup \{(0_{\mathcal{S}}, 0_{\mathcal{A}}), (0_{\mathcal{A}}, 0_{\mathcal{S}})\}$. G is called the *support graph*. A *$\mathcal{S}$-walk* in G is a sequence of vertices $(s_0, a_0, s_1, a_1, ..., s_k)$ for some $k \in \mathbb{N}$ with $s_i \in \mathcal{S}$ for all $0 \leq i \leq k$, $a_i \in \mathcal{A}$ for all $0 \leq i \leq k-1$, $(s_i, a_i) \in E$ for all $0 \leq i \leq k$, and $(a_{i-1}, s_i) \in E$ for all $1 \leq i \leq k$. k is called the *length* of the walk. We denote by W_k the set of all possible $\mathcal{S}$-walk of length k and $W := \cup_{k \in \mathbb{N}} W_k$. A *policy* Π is a function $\Pi : (k, w_k) \in \mathbb{N} \times W_k \mapsto \Pi_{k,w_k} \in \mathcal{M}_{=}(n,m)$ satisfying $\Pi_{k,w_k}(s,a) > 0 \implies J(s,a) = 1$ for all $s \in \{1,...,n\}$ and $a \in \{1,...,m\}$. We say that a policy is *deterministic* if Π_{k,W_k} is a 0/1 matrix for all k and w_k, it is *randomized* otherwise. If Π is constant, we say that the policy is *stationary*, otherwise it is called *history-dependent*. A policy Π induces a probability distribution over the (countable) set of all possible $\mathcal{S}$-walks. When Π is stationary, we often abuse notation and identify Π with a matrix.

We let $y_k^{\Pi} \in \mathbb{R}_+^n$ be the substochastic vector representing the state of the system in period k when following policy Π (from an initial distribution y_0^{Π}). That is $y_k^{\Pi}(i)$ is the probability of being in state i, for all $i = 1, ..., n$ at time k following policy Π. Similarly, we denote by $x_k^{\Pi} \in \mathbb{R}_+^m$ the substochastic vector representing the probability to perform action j, for all $j = 1, ..., m$, at time k following policy Π. By the law of total probability, we have $x_k^{\Pi} = \Pi^T \cdot y_k^{\Pi}$ for all $k \geq 0$. Similarly, given a stationary policy Π and an initial distribution y_0^{Π} at time 0, we have : $y_k^{\Pi} = P^T x_{k-1}^{\Pi} = P^T \cdot \Pi^T \cdot y_{k-1}^{\Pi}$ for all $k \geq 1$. Hence the state of the system at time $k \geq 0$ follows $y_k^{\Pi} = (P^T \cdot \Pi^T)^k \cdot y_0^{\Pi}$. The value $c^T x_k^{\Pi}$ represents the expected cost paid at time k following policy Π. One can define for each $i \in \mathcal{S} \setminus \{0\}$, $J_{\Pi}(i) := \limsup_{K \to +\infty} \sum_{k=0}^{K} c^T x_k^{\Pi}$ with $y_0^{\pi} := e_i$, and $J^*(i) := \min\{J_{\Pi}(i) : \Pi$ deterministic and stationary policy$\}$[3] (e_i is the characteristic vector of $\{i\}$ i.e. the 0/1 vector with $e_i(j) = 1$ iff $j = i$). Bertsekas and Tsistiklis [7] defined a stationary policy Π^* to be *optimal*[4] if $J^*(i) := J_{\Pi^*}(i)$ for all $i \in \mathcal{S} \setminus \{0\}$. They introduced the *Stochastic Shortest Path Problem* as the problem of finding such an optimal stationary policy.

Literature review

The stochastic shortest path problem is a special case of Markov Decision Process and it is also known as total reward undiscounted MDP [5, 6, 27]. It arises naturally in robot motion planning, from maneuvering a vehicle over unfamiliar terrain, steering a flexible needle through human tissue or guiding a swimming micro-robot through turbulent water for instance [2]. It has also many applications in operations research, artificial intelligence and economics: from inventory control, reinforcement learning to asset pricing (see for instance [32, 24, 3, 30]). SSP forms an important class of MDPs as it contains finite horizon MDPs, discounted MDPs (a euro tomorrow is worth less than a euro today) and average cost problems (through the so-called vanishing discounted factor approach) as special cases. It thus encapsulates most of the work on finite state/action MDPs. The stochastic shortest path problem was introduced first by Eaton and Zadeh in 1962 [11] in the context of pursuit-evasion games and it was later studied thoroughly by Bertsekas and Tsitsiklis [7].

MDPs were first introduced in the 50's by Bellman [4] and Shapley [28] and they have a long, rich and successful history (see for instance [5, 6, 27]). For most MDPs, it is known that there exists an optimal deterministic and stationary policy [27]. Building upon this fact, there are essentially three ways of solving such problems exactly (and some variants) : *value iteration* (VI), *policy iteration* (PI) and linear programming (LP). Value iteration and policy iteration are the original 50+ years old methods [4, 20]. The idea behind VI is to approximate the infinite horizon problem with a longer and longer finite one. The solution to the k-period approximation is built inductively from the optimal solution to the $(k-1)$-period problem using standard dynamic programing. The convergence of the method relies mainly on the theory

[2] If not we simply duplicate the actions.

[3] lim sup is used here as the limit need not be defined in general.

[4] note that it is not clear, a priori, that such a policy exists

of contraction mappings and Banach fixed-point theorem [5] for most MDPs. PI is an alternative method that starts from a feasible deterministic and stationary policy and iteratively improves the action in each state so as to converge to an optimal solution. It can be interpreted as a simplex algorithm where multiple pivots are performed in each step [21, 9]. As such it builds implicitly upon the geometry of the problem to find optimal solutions. Building explicitly upon this polyhedra, most MDPs can also be formulated as linear programs and as such they can thus be solved in (weakly) polynomial time [21, 10, 9, 19, 18].

In the context of the SSP, some hypothesis are required for standard methods and proof techniques to apply. Bertsekas and Tsitsiklis [7] introduced the notion of *proper* stationary policies : a stationary policy Π is said to be *proper* if $\mathbf{1}^T(P^T \cdot \Pi^T)^n \cdot e_i < 1$ for all $i = 1, ..., n$, that is, after n periods of time, the probability of reaching the target state is positive, from any initial state i. We say that such policies are *BT-proper* as we will introduce a slight generalization later. They proved that VI, PI and LP still work when two assumptions hold, namely, when (i) there exists a BT-proper policy and (ii) any BT-improper policy Π have at least one state i for which $J_\Pi(i) = +\infty$. In particular they show that one can restrict to deterministic policies. Their assumptions naturally discriminate between BT-proper and BT-improper policies. Exploiting further the discrepency between these policies, Bertsekas and Yu [8] showed that one can relax asumptions (i) and (ii) when the goal is to find an optimal *BT-proper* stationary policy. They could show that applying the standard VI and PI methods onto a perturbated problem where c is modified to $c + \delta \cdot \mathbf{1}$ with $\delta > 0$ and letting δ tends to zero over the iterations, yields an optimal BT-proper solution if (j) there exists a BT-proper policy and (jj) J^* is real-valued. Moreover they could also show that the problem can still be formulated (and thus solved) using linear programming, which settles the (weak) polynomiality of this extension. Some authors from the AI community proposed alternative extensions of the standard SSP introduced by Bertsekas and Tsitsiklis. It is easy to see that the most general one, entitled Stochastic and Safety Shortest Path problem [31], is a special case of Bertsekas and Yu's framework (it is a bi-objective problem that can be easily model in this framework using artificial actions of prohibited cost).

The question of whether SSP, in its original form or the later generalization by Bertsekas and Yu, can be solved in strongly polynomial time[5] is a major open problem for MDPs (see for instance [34]). It was proven in a series of breakthrough papers that it is the case for *fixed* discount rate (basically the same problem as before but where the transition matrix P is such that there is a fixed non-zero probability of ending up in 0 after taking any action). The result was first proved using interior point methods [33] and then the same author showed that the original policy iteration method proposed by Howard was actually strongly polynomial too [34] (the analysis was later improved [17]). The problem is still open for the undiscounted case but Policy Iteration is known to be exponential in that setting [15]. In contrast, value iteration was proved to be exponential even for the discounted case [14]. Because SSPs can be formulated as linear programs, the question relates very much to the existence of strongly polynomial time algorithms for linear programming, a very long-lasting open problem that was listed as one of the 18 mathematical problems of the 21st century by Smale in 1998 [29]. A possible line of attack is to study simplex-type of algorithms but existence of such algorithms is also a long standing open problem and relates to the Hirsch conjecture on the diameter of polyhedra. These questions are central in optimization, discrete geometry and computational complexity. Despite the fact that SSP exhibits strong additional properties over general LPs, these questions are still currently out of reach in this setting, too.

In practice, value iteration and policy iteration are the methods of choice when solving medium size MDPs. For large scale problems (i.e. most practical applications), approximate solutions are needed to provide satisfying solutions in a reasonable amount of time [26]. The field is known as Approximate Dynamic Programming and is a very active area of research. Most approximation methods are based on approximate versions of exact algorithms and developping new exact approaches is thus of great practical interest.

In this paper, we propose an extension of the frameworks of Bertsekas and Tsitsiklis [7] and Bertsekas and Yu [8]. We prove in section 3 that, in this setting, there is an optimal deterministic and stationary policy. Then we show in section 4 that the standard Value Iteration and Policy Iteration methods converge, and we give an alternative approach that generalizes Dijkstra's algorithm when the costs are non negative.

[5] a polynomial in the number of states and the number of actions

1.1 Notations and definitions

Given a directed graph $G(V, E)$, and a set $S \subset V$, we denote by $\delta^+(S)$ the set of arcs (u, v) with $u \in S$ and $v \notin S$, and by $N^+(S)$ the set of vertices $v \in V$ such that $(u, v) \in E$ for some $u \in S$. For convenience when S is a singleton, we denote $\delta^+(\{u\})$ by $\delta^+(u)$ and $N^+(\{u\})$ by $N^+(u)$. Then we define inductively $N_k^+(u) := N^+(N_{k-1}^+(u)) \setminus N_{k-1}^+(u) \cup \ldots \cup N_0^+(u))$ for $k \geq 1$ integer with $N_0^+(u) = \{u\}$. We denote by $R^+(u)$ the set of vertices *reachable* from u i.e. $R^+(u) = \bigcup_{k \geq 0} N_k^+(u)$. We can define $\delta^-(u)$, $N^-(u)$, $N_k^-(u)$ and $R^-(u)$ analogously. Clearly $v \in R^-(u)$ if and only if $u \in R^+(v)$. $R^-(u)$ are the vertices that can reach u. When confusion may arise, we denote $R^+(u)$ by $R_G^+(u)$ (and similarly for the other notations). We denote by $\mathbb{1}_A$ the indicator function associated with a set A i.e. $\mathbb{1}_A$ is a 0/1 function with $\mathbb{1}_A(x) = 1$ if and only if $x \in A$. For a vector $x \in \mathbb{R}^d$ and $I \subseteq \{1, ..., d\}$ we denote by $x[I]$ the restriction of x to the indices in I and $x(I) := \sum_{i \in I} x(i)$.

2 Our new framework

We start with a simple observation whose proof can be found in the Appendix (see section A.2).

Lemma 1. *For BT-proper stationary policies,* $\lim_{K \to +\infty} \sum_{k=0}^{K} x_k^{\Pi}$ *is finite for any initial state distribution* y_0^{Π}.

We now extend the notion of proper policies introduced by Bertsekas and Tsitsiklis using this alternative (relaxed) property and *from now on we will only use this new definition.*

Given a state $s \in \{1, ..., n\}$, a policy Π is said to be *s-proper* if $\sum_{k \geq 0} x_k^{\Pi}$ is finite, when $y_0^{\Pi} := e_s$. Observe that $\sum_{k \geq 0} y_k^{\Pi}$ is also finite for s-proper policies (as $y_k^{\Pi} = P^T x_{k-1}^{\Pi}$). In particular $\lim_{k \to +\infty} y_k^{\Pi} = 0$, that is the policy lead to the target state 0 with probability 1 from state s. The *s-stochastic-shortest-path problem* (s-SSP for short) is the problem of finding a s-proper policy Π of minimal cost $c^T \sum_{k \geq 0} x_k^{\Pi}$. We say that a policy is *proper* if it is s-proper for all s and *improper* otherwise. The *stochastic shortest path problem* (SSP) is the problem of finding a proper policy Π of minimal cost $c^T \sum_{k \geq 0} x_k^{\Pi}$ where $y_0^{\Pi} := \frac{1}{n}\mathbf{1}$. It is easily seen that the stochastic shortest path problem, as defined here, is also a special case of the s-SSP as one can add an artificial state with only one action that leads to all states in $\{1, ..., n\}$ with probability $\frac{1}{n}$. In the following two sections, unless otherwise stated, we restrict to the s-SSP. In this context, we often abuse notation and we simply call proper a s-proper policy.

Since for any policy Π (possibly history-dependent and randomized), Π_{k,w_k} are stochastic matrices, we have at any period $k \geq 0$, $\sum_{a \in \mathcal{A}(s)} x_k^{\Pi}(a) = y_k^{\Pi}(s)$. We also have $y_{k+1}^{\Pi}(s) = \sum_{a \in \mathcal{A}} p(s|a) x_k^{\Pi}(a)$ for all $s \in \{1, ..., n\}$. In matrix form this is equivalent to $y_k^{\Pi} = J^T x_k^{\Pi}$ and $y_{k+1}^{\Pi} = P^T x_k^{\Pi}$. This implies $J^T x_{k+1}^{\Pi} = P^T x_k^{\Pi}$ for all $k \geq 0$. We also have $J^T x_0^{\Pi} = e_s$. Now $x^{\Pi} := \sum_{k=0}^{\infty} x_k^{\Pi}$ is well-defined for proper policies. Summing up the previous relations over all periods $k \geq 0$ we get $(J - P)^T x^{\Pi} = e_s$. Hence the following linear program is a relaxation of the s-SSP problem[6] .

$$\begin{array}{lll} \min & & c^T x \\ (J-P)^T x & = & e_s \\ x & \geq & 0 \end{array} \qquad (P_s)$$

Observe that for a deterministic problem (i.e. when P is a 0/1 matrix), $(J - P)^T$ is the node-arc incidence matrix of a graph (up to a row) and the corresponding LP is the standard network flow

[6] We would like to stress on the fact that the LP relaxation we consider here is almost (except for the right hand side) the standard LP formulation of the problem of finding an optimal deterministic and stationary policy and it was already known for quite some time for many special cases of SSP (see [8] for instance). However while in the MDP community, the LP formulation comes as a corollary of other results, here we reverse the approach and introduce this formulation as a natural relaxation of the problem and we derive the standard results as (reasonably) simple corollaries. This is what allows to simplify, generalize and unify many results from the litterature. This is a simple yet major contribution of this paper. The notation and terminology is taken from [16]

relaxation of the deterministic shortest path problem. The vector x is sometimes called a network *flux* as it generalizes the notion of network flow.

We call a solution x to $(J-P)^T x = 0, x \geq 0$ a *transition cycle* and the cost of such a transition cycle x is $c^T x$. Negative cost transition cycles are the natural extension of negative cost cycles for deterministic problems (actually we could also consider only the extreme rays of $(J-P)^T x = 0, x \geq 0$; we defer the discussion to the journal version of the paper). One can check the existence of such objects by solving a linear program.

Lemma 2. *One can check in (weakly) polynomial time whether a stochastic shortest path instance admits a negative cost transition cycle through linear programming.*

We will prove in the sequel that the extreme points of $P_s := \{x \geq 0 : (J-P)^T x = e_s\}$ 'correspond' to proper deterministic and stationary policies. Hence, when the relaxation (P_s) has a finite optimum (i.e. when there is no transition cycle of negative cost and when a proper policy exists), this will allow to prove that, the s-SSP admits an optimal proper policy which is deterministic and stationary. This answers, for this problem, one fundamental question in Markov Decision Problem theory "Under what conditions is it optimal to restrict to deterministic and stationary policies ?" [27].

We can assume without loss of generality that there exists a path between all state node i and 0 in the support graph G. Indeed, if there is a node i with no path to 0 in G, then no s-proper policy will pass through i at any point in time (because then the probability of reaching the target state, starting from i, is zero, contradicting $\lim_{k\to+\infty} y_k^{\Pi} = 0$) ; we could thus remove i and the actions leading to i and iterate. It is easy to see that under this assumption, there is always a s-proper policy. Indeed the randomized and stationary policy Π that chooses an action in state i uniformly at random among $\mathcal{A}(i)$ will work : in this case, for each state i, there is in fact a non zero probability of choosing one of the paths from i to 0 after at most n periods of time.

Lemma 3. *Consider a s-SSP instance where there exists a path between all state node i and 0 in the support graph G. Then the policy that consists, for each state $i \in \mathcal{S}\setminus\{0\}$, in choosing uniformly at random an action in $\mathcal{A}(i)$ is a proper stationary policy.*

The discussion above also gives a simple algorithm for testing the existence of a proper policy for any instance of the SSP.

Lemma 4. *One can check in time $O(|U|\cdot(|U|+|V|+|E|))$ whether a s-SSP instance with support graph $G=(U,V,E)$ admits a proper policy or not.*

We are now ready to introduce the new assumptions that we will use to study the stochastic shortest path problem. They are the very natural extensions of the standard assumptions for the deterministic shortest path problem.

Assumption 1 *We consider s-SSP/SSP instances where :*

- *there exists a path between all state node i and 0 in the support graph G, and*
- *there is no negative cost transition cycle.*

As already observed, these assumptions can be checked in (weakly) polynomial time. Moreover, these assumptions implies that (P_s) has a finite optimum (from standard LP arguments). Also Bertsekas and Yu's framework is a special case of our setting as in the presence of negative cost transition cycles, $J^*(i)$ is not real-valued for some state i[7]. The main extension, with respect to Bertsekas and Yu, is that we allow for non-stationary proper policies in the first place.

[7] In order to prove this statement formally we shall prove that if $J^*(i)$ is real-valued for all i then there is no negative cost transition cycle : we can prove that when there exists a negative cost transition cycle, we can find one which is 'induced' by a (non proper) deterministic and stationary policy Π and that all vertices i on this cycle will have $J_{\Pi}(i) = -\infty$; for this, we need to extend our decomposition result to decompose transition cycles into extreme rays ; we leave the details for the journal version of the paper, but the proof of Proposition 14 gives the flavor of this latter result.

3 Existence of an optimal, deterministic and stationary policy

In this section, we will prove essential properties about $P_s := \{x \geq 0 : (J-P)^T x = e_s\}$. This will allow to prove that, under Assumption 1, we can restrict to optimal proper, deterministic and stationary policies. The following theorem can be seen as an extension of the *flow decomposition theorem* (see [1]).

Theorem 5. *Let $x \in \mathbb{R}^m$ be a feasible solution of (P_s). In strongly polynomial time, one can find $1 \leq k \leq m$, $x_1, ..., x_k, x_c \in \mathbb{R}^m$, and $\lambda_1, ..., \lambda_k \in [0,1]$ such that $x_1, ..., x_k$ are feasible solutions of (P_s), x_c satisfies $(J-P)^T x_c = 0, x_c \geq 0$, $\sum_{j=1}^k \lambda_j = 1$ and $x = \sum_{j=0}^k \lambda_j x_j + x_c$. Moreover, the vectors x_j are network flux corresponding to proper, deterministic and stationary policies, i.e. for all $j \in 1, \ldots, k$, there exists a proper, deterministic and stationary policy Π_j such that $x_j = x^{\Pi_j}$.*

Before we can prove this theorem, we need a couple of useful lemmas and definitions. Let $G = (U, V, E)$ and $x \in \mathbb{R}^m$ be a solution to $(J-P)^T x = e_s, x \geq 0$. Let G_x be the subgraph of G induced by the vertices in $\mathcal{S} \cup \mathcal{A}_x$ where $\mathcal{A}_x := \{a \in \{1, ..., m\}$ with $x(a) > 0$. G_x is called the *support graph of x in G*. We denote by E_x the set of edges of G_x.

Lemma 6. *There exists a path between all states reachable from s in G_x and $0_{\mathcal{S}}$. In other word, for all $i \in R^+_{G_x}(s)$, we have $i \in R^-_{G_x}(0_{\mathcal{S}})$.*

Proof. Let us define $\bar{x} \in \mathbb{R}^{|E_x|}$ as follows : $\bar{x}((s', a)) := x(a)$ for all $a \in A_x$ and s' the (unique) state with $a \in \mathcal{A}(s')$, and $\bar{x}((a, s')) := P(a, s') \cdot x(a)$ for all $a \in A_x$, and $s' \in S$ such that $P(a, s) > 0$. Observe that $\bar{x}$ is only defined on E_x and that $\bar{x} > 0$. Because x is a feasible solution to (P_s), $\bar{x}$ satisfies $\bar{x}(\delta^+_{G_x}(v)) - \bar{x}(\delta^-_{G_x}(v)) = \mathbb{1}_{\{s\}}(v) - \mathbb{1}_{\{0_{\mathcal{S}}\}}(v)$ for all $v \in G_x$ and $\bar{x} \geq 0$. It is thus a unit $(s, 0_{\mathcal{S}})$-flow in G_x. Now let us assume that there exists $i \in R^+_{G_x}(s)$ with $i \notin R^-_{G_x}(0)$. Summing up all flow constraints over $v \in R^+(i)$, we get $\bar{x}(\delta^+(R^+(i))) - \bar{x}(\delta^-(R^+(i))) = \mathbb{1}_{R^+(i)}(s)$ (we remove from now on the subscript G_x in order not to overload the notation). We have $\bar{x}(\delta^+(R^+(i))) = 0$ by definition of $R^+(i)$. But then $\bar{x}(\delta^-(R^+(i))) + \mathbb{1}_{R^+(i)}(s) = 0$. Since $\bar{x}(\delta^-(R^+(i))) \geq 0$, this implies $s \notin R^+(i)$ and $\bar{x}(\delta^-(R^+(i))) = 0$. Now because $s \notin R^+(i)$ and $s \in R^-(i)$ (by hypothesis), there is at least one arc of E_x in $\delta^-(R^+(i))$ but this implies $\bar{x}(\delta^-(R^+(i))) > 0$ as $\bar{x} > 0$, a contradiction.

Given a proper, deterministic and stationary policy Π, we denote by G_Π the subgraph of G induced by the state vertices in $\mathcal{S}$ and the actions vertices in Π. Now let G^s_Π be the subgraph of G_Π induced by the vertices in $R^+(s)$. G^s_Π is called the *support graph* of Π (it is easily seen that it corresponds to the subgraph induced by the states and actions that we might visit under policy Π). Because Π is proper, 0_S is reachable from each state i in G^s_Π. Let us denote by $\mathcal{S}'$ the state vertices in G^s_Π and $\Pi(\mathcal{S}')$ the actions associated with $\mathcal{S}'$ in Π. We also denote by $P_{\mathcal{S}'}$ the restriction of P to the rows in $\mathcal{S}'$ and the columns in $\Pi(\mathcal{S}')$ ($P_{\mathcal{S}'}$ is a $|\mathcal{S}'| \times |\mathcal{S}'|$ matrix). Following the same arguments as in Section A.2, $\lim_{k \to +\infty}(P_{\mathcal{S}'})^k = 0$ and thus $(I_{\mathcal{S}'} - P_{\mathcal{S}'})$ is invertible. Now observe that $(I_{\mathcal{S}'} - P_{\mathcal{S}'})^T x^\Pi[\Pi(S')] = e'_s$ for $x^\Pi := \sum_{k=0}^{+\infty} x^\Pi_k$, with $y^\Pi_0 := e_s$ (e'_s is the restriction of e_s to the indices in $\mathcal{S}'$). Indeed $x^\Pi(a) = 0$ for all $a \notin \Pi(S')$ and thus $(I_{\mathcal{S}'} - P_{\mathcal{S}'})^T x^\Pi[\Pi(S')] = e'_s$ corresponds to the constraints of (P_s) associated with the rows in $\mathcal{S}'$. We thus have the following lemma.

Lemma 7. *Given a proper, deterministic and stationary policy Π, the flux vector x^Π associated with Π and defined by $x^\Pi := \sum_{k=0}^{+\infty} x^\Pi_k$, with $y^\Pi_0 := e_s$ satisfies $x^\Pi[\Pi(S')] = (I_{\mathcal{S}'} - P_{\mathcal{S}'})^{-T} e'_s$ and $x^\Pi(a) = 0$ for all $a \notin \Pi(\mathcal{S}')$, with $\mathcal{S}'$,$\Pi(\mathcal{S}')$, $I_{\mathcal{S}'}$,$P_{\mathcal{S}'}$ and e'_s defined as above.*

The following Lemma is easy to prove using similar flow arguments as in the proof of Lemma 6.

Lemma 8. *Let Π be a proper, deterministic and stationary policy. We have $G^s_\Pi = G_{x^\Pi}$. Moreover if $x \in P_s$ and $\Pi(S) \subseteq \mathcal{A}_x$, then G^s_Π is a subgraph of G_x and $x^\Pi(a) \geq x(a)$ for some $a \in \mathcal{A}_x$.*

Before proving Theorem 10, we need a final Lemma.

Lemma 9. *Let $G = (U, V, E)$ be the support graph of a s-SSP instance and assume that there is a path from every state vertex i to $0_{\mathcal{S}}$ in G. Then in time $O(|U|+|V|+|E|)$, one can find a proper, deterministic and stationary policy Π.*

Proof. We know that, $0 \in R^+(i)$ for all i, is enough to ensure that there is a proper policy by Lemma 3. Now if there exists a state vertex i in G with $|\mathcal{A}(i)| > 1$, we can delete from G an action in $\mathcal{A}(i)$ that does not remove 0 from $R^+(i)$. Such an action exists as it is enough to keep an action $a \in A(i)$ with minimum distance to 0 (in terms of arc) to ensure that 0 is still in $R^+(i)$ after deletion (by minimality of the distance to 0, such an action has a directed path to 0 that does not go through i). If $|\mathcal{A}(i)| = 1$ for all i then the only possible policy is proper (from Lemma 3), deterministic and stationary. We can implement such a procedure in time $O(|U| + |V| + |E|)$ by computing $N_k^-(0)$ for all $k \leq |U| + |V|$ and a 0-anti-arborescence A using a breadth first search algorithm : we then keep only the actions in A.

We are now ready to prove the main Theorem of this section.

Theorem 10. *Let $x \in \mathbb{R}^m$ be a feasible solution of (P_s). In strongly polynomial time, one can find $1 \leq k \leq m$, $x_1, ..., x_k, x_c \in \mathbb{R}^m$, and $\lambda_1, ..., \lambda_k \in [0, 1]$ such that $x_1, ..., x_k$ are feasible solutions of (P_s), x_c satisfies $(J - P)^T x_c = 0, x_c \geq 0$, $\sum_{j=1}^k \lambda_j = 1$ and $x = \sum_{j=0}^k \lambda_j x_j + x_c$. Moreover, the vectors x_j are network flux corresponding to proper, deterministic and stationary policies, i.e. for all $j \in 1, \ldots, k$, there exists a proper, deterministic and stationary policy Π_j such that $x_j = x^{\Pi_j}$.*

Proof. We prove first that such a decomposition exists for any $x \in P_s$. Let x be a smallest counter-example (in terms of $|A_x|$). Because x is a feasible solution of (P_s), we know by Lemma 6 that there exists a path between all states reachable from s in G_x and 0. Now from Lemma 9, we know that there exists a proper, deterministic and stationary policy Π to which we can associate and compute a flux x^Π using Lemma 7. Let $\lambda \geq 0$ be the maximum value such that $x' := x - \lambda x^\Pi \geq 0$. By Lemma 8 we have that G_{x^Π} is a subgraph of G_x and thus $\lambda > 0$ (as $x > 0$ on $\mathcal{A}_x$). We also have $\lambda \leq 1$ by the same Lemma. Moreover by maximality of λ, there is an arc $a \in \mathcal{A}_x$ such that $x(a) > 0$ and $x'(a) = 0$. Hence $\mathcal{A}_{x'} \subset \mathcal{A}_x$. If $\lambda = 1$, x' is a solution to $(J - P)^T x = 0, x \geq 0$ and $x := x^\Pi + x'$ provides a decomposition for x, a contradiction. Else, $\frac{1}{1-\lambda} x'$ is a solution to (P_s) with $|\mathcal{A}_{x'}| < |\mathcal{A}_x|$. By minimality of the counter-example, we can assume that there exists a decomposition for $\frac{1}{1-\lambda} x'$. Now we can get a decomposition for x from the decomposition for $\frac{1}{1-\lambda} x'$ by scaling the multipliers by 1-λ and using x^Π with multiplier λ, this is contradiction. Clearly, we can make the proof algorithmic and because $A_{x'} \subset A_x$ at each iteration, the algorithm will terminate in at most $|A_x|$ steps with a set of $k \leq |A_x|$ solutions $x_1, ..., x_k$ to (P_s) and a vector x_c satisfying the theorem.

Corollary 11. *Under Assumption 1, the s-SSP admits an optimal proper, deterministic and stationary policy.*

Proof. We know from linear programming that when a LP has finite optimum, we can find an optimal solution in an extreme point. For (P_s) this is guaranteed by Assumption 1. But an extreme point x of P_s cannot be expressed as a convex combination of other points of P_s by definition. As such, using Theorem 10, x must be equal to x^Π for some proper, deterministic and stationary policy Π. Now $c^T x^\Pi$ is precisely the cost of policy Π. Hence we have a feasible solution to our original problem which is optimal for the linear relaxation (P_s). It is thus optimal for the original problem.

We can deduce from what preceeds a result which is standard for the deterministic shortest path problem : *Bellman optimality conditions.*

Lemma 12. *Let Π be an optimal proper, deterministic and stationary solution to the s-SSP (under Assumption 1). Let G^s_Π be the support graph of Π. For all state vertex i in G^s_Π, Π is optimal for i-SSP.*

Proof. Observe first that i-SSP satisfies Assumption 1. Now suppose Π is not optimal for i-SSP. We know from Corollary 11 that i-SSP admits an optimal proper, deterministic and stationary policy Π_i. Now the (history-dependent and non stationary) policy Π' that consists in applying policy Π to problem s-SSP, up to when state i is reached (if it ever is) and then applying policy Π_i is a proper policy. The value of this policy is better than the value of Π as there exists a realization where i is reached, a contradiction.

4 Algorithms

We focussed, up to now and without loss of generality, on the s-SSP problem. Bellman optimality conditions (i.e. Lemma 12) also tells us that, under Assumption 1, we can actually restrict attention to the

SSP problem as well without loss of generality. Indeed we already observed that SSP can be converted to a s-SSP problem by simply adding an artificial state s and a unique action available from s that lead to all states $i = 1, ..., n$ with probability $\frac{1}{n}$. Now there is a one-to-one correspondance between the policies of SSP and the policies of the auxiliary s-SSP problem and hence any proper, deterministic and stationary solution Π to SSP is optimal if and only if it is optimal for the auxiliary problem. But by Lemma 12, an optimal policy Π^* for SSP is optimal for i-SSP for all $i = 1, ..., n$ (as all i are in G'_{Π^*}). It is easy to see that all theorems from the previous section extend naturally to the SSP setting. Of course, some definitions and results have to be slightly adapted : for instance, the flux vector x^Π associated with a proper deterministic and stationary policy is now $x^\Pi := \sum_{k=0}^{+\infty} x_k^\Pi$ with $y_0^\Pi := \frac{1}{n}\mathbf{1}$ and it satisfies $x^\Pi = (I - P_\Pi)^{-T}\frac{1}{n}\mathbf{1}$ (see Lemma 7 for the previous relation), where P_Π is the $n \times n$ matrix obtained from P by keeping only the rows corresponding to actions in Π. For algorithmic reasons, it is more convenient to deal with the SSP problem as there is no problem of degeneracy : the feasible basic solution x^Π (it is indeed now the basic solution associated with the basis $(I - P_\Pi)^T$) has positive values on the actions in Π. In this section, we will therefore focus on the SSP problem. The corresponding linear programming formulation is (in principle, the right hand side should be $\frac{1}{n}\mathbf{1}$ but we simply rescaled it):

$$\begin{array}{ll} \min & c^T x \\ (J - P)^T x & = \mathbf{1} \\ x & \geq 0 \end{array} \qquad (P)$$

One possible way of solving the previous model is to use any polynomial time algorithm for linear programming. This would lead to weakly polynomial time algorithms for SSP. As pointed out in the introduction, there are two standard alternatives for solving a MDP : Value Iteration and Policy Iteration. We prove in the next two sections the convergence of these methods under Assumption 1. Then we give another new iterative method based on the standard primal-dual approach to linear programming : this can be considered as a natural generalization of Dijkstra's algorithm.

4.1 Value Iteration

We denote by $\mathcal{P}$ the set of all proper policies for SSP. For all $i = 1, ..., n$, we define $V^*(i)$ to be the optimal value of (P_i) (again under Assumptions 1), i.e. $V^*(i) := \min_{\Pi \in \mathcal{P}} c^T x^\Pi$ with $y_0^\Pi = e_i$. This is refered to as the *value* of state i. We have in particular $V^*(i) = \min_{\Pi \in \mathcal{P}} \lim_{K \to +\infty} \sum_{k=0}^{K} c^T x_k^\Pi$ by definition of x_k^Π. In the following, we show that we can switch the min and lim operators with some care. We need first to introduce an auxiliary SSP instance obtained from $(\mathcal{S}, \mathcal{A}, J, P, c)$ by adding an action of cost $M(i)$ for each state $i = 1, ..., n$ that lead to state 0 with probability one, with $M(i)$ "big enough" . We call aux-SSP this auxiliary problem. Observe that in aux-SSP, there are proper policies that terminate in at most k time periods for all $k \geq 1$, from any starting state. Indeed one can always chose an auxiliary action in period $k - 1$. Let us denote by $\mathcal{P}^k$ the proper policies in aux-SSP that terminate in at most k steps and by $\mathcal{P}_{aux}$ the proper policies for aux-SSP. Observe that $V_K(i) := \min_{\Pi \in \mathcal{P}^K} \sum_{k=0}^{K} c^T x_k^\Pi$ is well-defined for each $K \geq 1$. In fact it is easy to prove by induction that it follows the dynamic programming formula : $V_k(i) = \min\{V_{k-1}(i), \min_{a \in \mathcal{A}(i)} c(a) + \sum_j p(j|a) V_{k-1}(j)\}$ for all $k \geq 2$ and $V_1(i) = M(i)$ for all $i = 1, ..., n$ (an optimal, deterministic non-stationary policy Π_K^* can be recovered easily too) : $V_k(i)$ is indeed the optimal value starting from i among policies in $\mathcal{P}^k$. The following result can be seen as an extension of Bellman-Ford algorithm for the deterministic shortest path problem. We give the proof in Section A.3.

Theorem 13. *For all $i = 1, ..., n$, if $M(i) \geq V^*(i)$, then we have $V^*(i) = \lim_{K \to +\infty} V_K(i)$.*

Notice that it is easy to find initial values for $M(i)$ satisfying the previous Theorem. Indeed one can use $V^\Pi(i)$, the values for state i when using policy Π for any i-proper policy Π. We can actually easily find a proper deterministic and stationary policy for SSP (i.e. for all i simultaneously) by extending Lemma 9 to SSP.

The algorithm that consists in evaluating V_k iteratively until $||V_k - V_{k-1}||_\infty$ is below some threshold is called *Value Iteration*. Value Iteration was already known to converge for SSP in the presence of transition cycles of cost zero, when initialized appropriately, see Bertsekas and Yu [8].

We now explain how to recover an optimal proper, deterministic and stationary policy given the optimal vector V^*. Let us consider the dual linear program (D) of (P) :

$$\begin{array}{ll} \max & \mathbf{1}^T y \\ (J-P)y \leq c & \end{array} \tag{D}$$

By definition of $V^*(i)$ and by Lemma 12, we know that V^* satisfies $V^*(i) = \min_{a \in \mathcal{A}(i)} c(a) + \sum_j p(j|a)V^*(j)$ for all $i = 1, ..., n$. Also extending Corollary 11 to SSP, we know that there exists an optimal proper deterministic and stationary policy Π^* with $V^*(i) = c(\Pi^*(i)) + \sum_j p(j|\Pi^*(i))V^*(j)$ for all $i = 1, .., n$. In particular, $y^* := V^*$ is feasible for (D) and because the pair (x^{Π^*}, y^*) satisfies the complementary slackness conditions, y^* is optimal for (D).

Now let us reverse the complementary slackness conditions. An optimal solution x^* to (P) can have $x^*(a) > 0$ only if $V^*(i) = c(a) + \sum_j p(j|a)V^*(j)$. Let $\mathcal{A}^*$ be the set of all such actions and let us restrict our instance of SSP to those actions in $\mathcal{A}^*$. Because there is an optimal proper, deterministic and stationary policy Π^* for SSP and because such a policy must use only actions in $\mathcal{A}^*$, we know that there is a path from every state to the target state 0 in the support graph $G^* = (U^*, V^*, E^*)$ of this instance. Now the stationary policy Π' consisting in choosing uniformly at random an action in $\mathcal{A}^*(i)$ for each state i is proper by Lemma 3 and $x^{\Pi'} := \lim_{K \to +\infty} \sum_{k=0}^{K} x_k^{\Pi'}$ is feasible for (P). Observe that by construction, the pair $(x^{\Pi'}, y^*)$ satisfies the complementary slackness conditions and thus $x^{\Pi'}$ is also optimal for (P). Extending theorem 10 to SSP and by optimality of $x^{\Pi'}$, we can decompose $x^{\Pi'}$ into a convex combination of vectors x^{Π_j} associated with proper, deterministic and stationary policies Π_j. Again by optimality of $x^{\Pi'}$, x^{Π_j} are also optimal for (P). We can thus use Theorem 10 to get an optimal, proper, deterministic and stationary policy for the problem. In fact we can stop after the first application of Lemma 9 and thus get such a policy in time $O(|U^*| + |V^*| + |E^*|)$.

N.B. We can define an approximate proper solution Π_k at each step k of Value Iteration by considering an approximate version of the complementary slackness theorem. We defer the discussion to the journal version of the paper.

4.2 Policy Iteration

An alternative to Value Iteration is to use a simplex algorithm to solve (P). In order to do so we need an initial basis. We can use Lemma 9 to find a proper deterministic and stationary policy Π. Then as we already observed, $x^{\Pi} = (I - P_{\Pi})^{-T}\mathbf{1}$ is a non-degenerate feasible basic solution of (P). Because the basic solutions are non-degenerate, we can implement any pivot rule from this initial basic solution and the simplex algorithm will converge in a finite number of steps. This type of algorithm is often referred to as *simple policy iteration* in the litterature. This proves that simple PI terminates in a finite number of steps. Unfortunately, most pivot rules are known to be exponential in n and m in the worst case [23].

In contrast with simple policy iteration, Howard's original policy iteration method [20] changes the actions of a (basic) policy in each state i for which there is an action in $\mathcal{A}(i)$ with negative *reduced cost*. We will prove now that this method converges under Assumptions 1. For this, we will prove that the method iterates over proper deterministic and stationary policies and that the cost is decreasing at each iteration. Given a proper deterministic and stationary policy Π, $x^{\Pi} = (I - P_{\Pi})^{-T}\mathbf{1}$ is the basic feasible solution of (P) associated with the basis $(I - P_{\Pi})^T$. We define the *reduced cost* vector associated with c and Π as $\bar{c}^{\Pi} := c - c_{\Pi}(I - P_{\Pi})^{-T}(J - P)^T$ following linear programming (in order not to overload the notations we consider c as a row vector in this section). Let us denote by $\mathcal{A}^{>}(\Pi)$ the set of actions a of $\mathcal{A}$ such that $\bar{c}^{\Pi}(a) < 0$. We know from linear programming that if $\bar{c}^{\Pi}(a) \geq 0$ for all a, then x^{Π} (and thus Π) is optimal. If $\mathcal{A}^{>}(\Pi) \neq \emptyset$, then we can swap actions in Π with actions in $\mathcal{A}^{>}(\Pi)$ for each state where such an action exists. Let us denote by Π' the resulting policy. The proof of the following proposition is given in Appendix A.4.

Proposition 14. *Π' is proper and $c \cdot x^{\Pi'} < c \cdot x^{\Pi}$*

Because we have a finite number of proper deterministic and stationary policy, we can conclude that Howard's policy iteration algorithm converges in a finite number of steps.

Theorem 15. *Under Assumption 1, Howard's PI method converges in a finite number of steps.*

Observe that it is important not to change actions which are not strictly improving. Indeed, in this case it is easy to build deterministic examples where Lemma 14 fails (see for instance Fig. 2 in Section A.1). As for value iteration, prior to this work policy iteration was not known to converge in this setting. And again, as for VI, unfortunately Howard's Policy Iteration can be exponential in n and m [13].

4.3 The Primal-Dual algorithm : a generalization of Dijkstra's algorithm

Primal-dual algorithms proved very powerful in the design of efficient (exact or approximation) algorithms in combinatorial optimization. Edmonds' algorithm for the weighted matching problem [12] is probably the most celebrated example. It is well-known that for the deterministic shortest path problem, when the costs are non negative, the primal-dual approach corresponds to Dijkstra's algorithm [25]. We extend this approach to the SSP setting. Let us first recall the linear formulation of the problem and its dual :

$$\begin{array}{lll} \min & c^T x & \\ (J-P)^T x & = \mathbf{1} & \text{(P)} \\ x & \geq 0 & \end{array} \qquad\qquad \begin{array}{lll} \max & \mathbf{1}^T y & \\ (J-P)\ y & \leq c & \text{(D)} \end{array}$$

The primal-dual algorithm works as follows here. Consider a feasible solution $\bar{y}$ to (D) (initially $\bar{y} = 0$ is feasible if $c \geq 0$). Now let $\bar{\mathcal{A}} := \{a \in \mathcal{A} : \mathbf{1}_a^T\ (J-P)\ y = c_a\}$. We know from complementary slackness that $\bar{y}$ is optimal if and only if there exists $x \geq 0 : (J-P)^T x = \mathbf{1}$ and $x_a = 0, \forall a \notin \bar{\mathcal{A}}$ ((P) admits an finite optimum by Assumption 1). The problem can be rephrased as a so-called restricted primal (RP), where $J_{\bar{\mathcal{A}}}, P_{\bar{\mathcal{A}}}$ and $x_{\bar{\mathcal{A}}}$ are the restrictions of J, P, x to the row in $\bar{\mathcal{A}}$. We also give its corresponding dual problem (DRP).

$$\begin{array}{lll} \min & \mathbf{1}^T z & \\ (J_{\bar{\mathcal{A}}} - P_{\bar{\mathcal{A}}})^T x_{\bar{\mathcal{A}}} + z & = \mathbf{1} & \text{(RP)} \\ x_{\bar{\mathcal{A}}}, z & \geq 0 & \end{array} \qquad\qquad \begin{array}{lll} \max & \mathbf{1}^T y & \\ (J_{\bar{\mathcal{A}}} - P_{\bar{\mathcal{A}}})\ y & \leq 0 & \text{(DRP)} \\ y & \leq \mathbf{1} & \end{array}$$

If there is a solution of cost 0 to (RP) then we have found an optimal solution to our original problem. Else, we use an optimal, positive cost solution $\underline{y}$ to (DRP) and we update the initial solution by setting $\bar{y} := \bar{y} + \epsilon \underline{y}$ with $\epsilon \geq 0$ maximum with the property that $\bar{y} + \epsilon \underline{y}$ remains feasible for (D), and we iterate. The algorithm is known to converge in a finite number of steps ((RP) being non degenerate, no anti-cycling rule is needed to guarantee finiteness here [25]) and this provides an alternative approach to the problem as long as we can also solve (RP) and (DRP).

Observe that (RP) can be interpreted as a SSP problem with action set $\bar{\mathcal{A}} \cup \{m+1, ..., m+n\}$, where actions $m+k$, for all $k = 1, ..., n$ is an artificial action associated with state k that lead to the target state 0 with probability one. The cost of actions in $\bar{\mathcal{A}}$ is zero while the cost of the artificial actions $m+1, ..., m+n$ is one. The primal-dual approach thus reduces the initial problem to a sequence of simpler 0/1 cost SSP problems. Note that (RP) is actually the problem of maximizing the probability of reaching state 0 using only actions in $\bar{\mathcal{A}}$. This problem is known in the AI community as MAXPROB [22]. Little is known about this problem. We know though that it can be solved in weakly polynomial time because it fits into our framework and we can thus solve it using linear programming. We could also use Value Iteration, the simplex method or Policy Iteration as described in the previous subsections. Some simplex rules are known to be exponential in this setting [23] : the question of the existence of a strongly polynomial algorithm is thus wide open for this subproblem too and we believe that MAXPROB deserves attention on its own. Using Howard's policy iteration algorithm to solve the auxiliary problem, the primal-dual approach provides an alternative finite algorithm to solve SSP for non negative costs instances.

Theorem 16. *When $c \geq 0$, the primal-dual algorithm can be initialized with $\bar{y} = 0$ and if the MAXPROB subproblems are solved using Howard's Policy Iteration (or any other simple Policy Iteration method), then it terminates in a finite number of steps.*

We are investigating the complexity of this extension of Dijkstra's algorithm to the SSP. Observe that we do not need to impose that c is non negative to apply the primal-dual approach. In fact, one can use

the standard trick of adding an artificial constraint $\sum_a x_a \leq M$ to the problem, with M "big" to find an initial dual solution and iterate the algorithm [25]. The structure of the subproblem changes but it can still be solved using the simplex method. This provides an alternative approach to Value Iteration and Policy Iteration in the general case too.

References

1. R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.
2. R. Alterovitz, T. Simon, and K. Goldberg. The stochastic motion roadmap: A sampling framework for planning with markov motion uncertainty. In M. P. W. Burgard et al. (Eds.), editor, *Robotics: Science and Systems III (Proc. RSS 2007)*, pages 233–241, 2008.
3. N. Bäuerle and U. Rieder. *Markov Decision Processes with Applications to Finance: Markov Decision Processes with Applications to Finance*. Springer Science & Business Media, 2011.
4. R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition, 1957.
5. D. P. Bertsekas. *Dynamic programming and optimal control. Volume I*. Athena Scientific optimization and computation series. Belmont, Mass. Athena Scientific, 2005.
6. D. P. Bertsekas. *Dynamic programming and optimal control. Volume II*. Athena Scientific optimization and computation series. Belmont, Mass. Athena Scientific, 2012.
7. D. P. Bertsekas and J. N. Tsitsiklis. An analysis of stochastic shortest path problems. *Math. Oper. Res.*, 16(3):580–595, Aug. 1991.
8. D. P. Bertsekas and H. Yu. Stochastic shortest path problems under weak conditions, 2016.
9. E. V. Denardo. On Linear Programming in a Markov Decision Problem. *Management Science*, 16(5):281–288, 1970.
10. F. d'Epenoux. A probabilistic production and inventory problem. *Management Science*, 10(1):98–108, 1963.
11. J. H. Eaton and L. A. Zadeh. Optimal pursuit strategies in discrete-state probabilistic systems. *J. Basic Eng.*, 84(1):23–29, Mar. 1962.
12. J. Edmonds. Maximum matching and a polyhedron with (0,1) vertices. *J. Res. Nat. Bur. Standards*, 69:125–130, 1965.
13. J. Fearnley. Exponential lower bounds for policy iteration. In S. Abramsky, C. Gavoille, C. Kirchner, F. Meyer auf der Heide, and P. Spirakis, editors, *Automata, Languages and Programming*, volume 6199 of *Lecture Notes in Computer Science*, pages 551–562. Springer Berlin Heidelberg, 2010.
14. E. A. Feinberg and J. Huang. The value iteration algorithm is not strongly polynomial for discounted dynamic programming. *Oper. Res. Lett.*, 42(2):130–131, Mar. 2014.
15. O. Friedmann. An exponential lower bound for the parity game strategy improvement algorithm as we know it. In *Proceedings of the 24th LICS*, pages 145–156, 2009.
16. T. D. Hansen. *Worst-case Analysis of Strategy Iteration and the Simplex Method*. PhD thesis, Aarhus University, 2012.
17. T. D. Hansen, P. B. Miltersen, and U. Zwick. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *J. ACM*, 60(1):1:1–1:16, Feb. 2013.
18. O. Hernández-Lerma and J.-B. Lasserre. The linear programming approach. In E. Feinberg and A. Shwartz, editors, *Handbook of Markov Decision Processes*, volume 40 of *International Series in Operations Research & Management Science*, pages 377–407. Springer US, 2002.
19. A. Hordijk and L. C. M. Kallenberg. Linear programming and markov decision chains. *Management Science*, 25(4):352–362, 1979.
20. R. A. Howard. *Dynamic programming and Markov processes*. The MIT press, New York London, Cambridge, MA, 1960.
21. A. S. Manne. Linear programming and sequential decisions. *Management Science*, 6(3):259–267, 1960.
22. Mausam and A. Kolobov. *Planning with Markov Decision Processes: An AI Perspective*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012.
23. M. Melekopoglou and A. Condon. On the complexity of the policy improvement algorithm for markov decision processes. *ORSA Journal on Computing*, 6(2):188–192, 1994.
24. R. Merton. An intertemporal capital asset pricing model. *Econometrica*, 41(5):867–887, 1973.
25. C. Papadimitriou and K. Steiglitz. *Combinatorial optimization: Algorithms and complexity*. Prentice-Hall, 1982.
26. W. B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality (Wiley Series in Probability and Statistics)*. Wiley-Interscience, 2007.
27. M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.
28. L. S. Shapley. Stochastic games. *Proceedings of National Academy of Science*, 39(10):1095–1100, 1953.

29. S. Smale. Mathematical problems for the next century. *The Mathematical Intelligencer*, 20(2):7–15, 1998.
30. R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.
31. F. Teichteil-Königsbuch. Stochastic safest and shortest path problems. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pages 1825–1831. AAAI Press, 2012.
32. D. J. White. A survey of applications of markov decision processes. *The Journal of the Operational Research Society*, 44(11):1073–1096, 1993.
33. Y. Ye. A new complexity result on solving the markov decision problem. *Mathematics of Operations Research*, 30(3):733–749, 2005.
34. Y. Ye. The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, 36(4):593–603, 2011.

A Appendix

A.1 An example

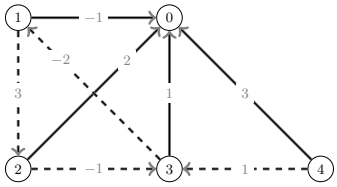

Fig. 2. A deterministic shortest path (with target state 0) : the dark "actions" represent the current policy, and the dashed "actions" have non positive reduced cost ; changing all actions with non positive reduced cost yield a new policy which is not proper.

A.2 Proof of Lemma 1

Proof. $x_k^{\Pi} = \Pi^T \cdot P^T \cdot x_{k-1}^{\Pi}$ for all $k \geq 1$ and $x_0^{\Pi} = \Pi^T y_0^{\Pi}$. Therefore $x_k^{\Pi} = (\Pi^T \cdot P^T)^k \cdot \Pi^T y_0^{\Pi}$, where y_0^{Π} is the original state distribution. It follows that $\sum_{k=0}^{K} x_k^{\Pi} = \sum_{k=0}^{K}((\Pi^T \cdot P^T)^k \cdot \Pi^T y_0) = (\sum_{k=0}^{K}(\Pi^T \cdot P^T)^k) \cdot \Pi^T y_0$ and because of the standard Lemma 17, it implies that $I - \Pi^T \cdot P^T$ is invertible and that $\lim_{K\to+\infty} \sum_{k=0}^{K} x_k^{\Pi} = (I - \Pi^T \cdot P^T)^{-1} \cdot \Pi^T y_0$. ($\lim_{k\to+\infty} \Pi^T \cdot P^T = 0$ by definition of BT-properness since $\mathbf{1}^T(P^T \cdot \Pi^T)^n \cdot e_i < 1$ for all $i = 1, ..., n$).

Lemma 17. *Let Q be a matrix with $\lim_{k\to+\infty} Q^k = 0$. Then $I - Q$ is invertible, $\sum_{k\geq 0} Q^k$ is well defined and $\sum_{k\geq 0} Q^k = (I - Q)^{-1}$.*

A.3 Proof of Theorem 13

We will prove that $\min_{\Pi\in\mathcal{P}} \lim_{K\to+\infty} \sum_{k=0}^{K} c^T x_k^{\Pi} = \lim_{K\to+\infty} \min_{\Pi\in\mathcal{P}^K} \sum_{k=0}^{K} c^T x_k^{\Pi}$ with $y_0^{\Pi} := e_i$, for all $i = 1, ..., n$, by proving both inequalities.

$\boxed{\leq}$ Let Π_K^* be an optimal solution to $\min_{\Pi\in\mathcal{P}^K} \sum_{k=0}^{K} c^T x_k^{\Pi}$ computed by dynamic programming (as described above). Π_K^* is a proper policy for aux-SSP for all K. By feasibility of Π_K^*, we thus have $V_K(i) = c^T \sum_{k=0}^{K} x_k^{\Pi_K^*} \geq \min_{\Pi\in\mathcal{P}_{aux}} \lim_{K\to+\infty} \sum_{k=0}^{K} c^T x_k^{\Pi}$ (observe that this minimum is well defined since we are still satisfying Assumptions 1 in aux-SSP). By construction $\{V_K(i), K \geq 1\}$ is non-increasing, hence because it is bounded from below, it converges and $\lim_{K\to+\infty} V_K(i)$ is well-defined. Taking the limit we get $\lim_{K\to+\infty} c^T \sum_{k=0}^{K} x_k^{\Pi_K^*} \geq \min_{\Pi\in\mathcal{P}_{aux}} \lim_{K\to+\infty} \sum_{k=0}^{K} c^T x_k^{\Pi}$. But $\min_{\Pi\in\mathcal{P}_{aux}} \lim_{K\to+\infty} \sum_{k=0}^{K} c^T x_k^{\Pi} \geq \min_{\Pi\in\mathcal{P}} \lim_{K\to+\infty} \sum_{k=0}^{K} c^T x_k^{\Pi}$ if $M(i)$ is chosen so that auxiliary actions can be assumed not to be used in an optimal policy in $\mathcal{P}_{aux}$. This is the case for $M(i) \geq V^*(i)$.

$\boxed{\geq}$ Let Π^* be an optimal proper deterministic and stationary solution to $\min_{\Pi\in\mathcal{P}} \lim_{K\to+\infty} \sum_{k=0}^{K} c^T x_k^{\Pi}$ (Π^* exists in our setting by Corollary 11). Let us denote by $\bar{\Pi}$ the policy of $\mathcal{P}_{aux}$ that chooses the auxiliary action for each state. Consider the policy Π_K of $\mathcal{P}^K$ obtained from using Π^* in periods $0, ..., K-1$ and policy $\bar{\Pi}$ in period K. By feasibility of Π_K, we have $c^T x_K^{\Pi_K} + \sum_{k=0}^{K-1} c^T x_k^{\Pi_K} \geq \min_{\Pi\in\mathcal{P}^K} \sum_{k=0}^{K} c^T x_k^{\Pi}$. Now taking the limit as K tends to infinity, we have the result since $\lim_{K\to+\infty} x_K^{\Pi_K} = \lim_{K\to+\infty} x_K^{\Pi^*} = 0$ as Π_K differs from Π^* only in period K, and Π^* is i-proper.

A.4 Proof of Proposition 14

Proof. We denote by y^{Π} the dual solution associated with Π i.e. $y^{\Pi} = c_{\Pi}(I - P_{\Pi})^{-T}$. Assume for contradiction that Π' is not proper. Let $G_{\Pi'}$ be the support graph of this policy. Since Π' is not proper, there exists a non empty set of states that are not in $R^-(0)$. It implies that there is a set of vertices V in $G_{\Pi'}$ such that $0_{\mathcal{S}}, 0_{\mathcal{A}} \notin V$ and $\delta^+(V) = \emptyset$. Now there exists an action a of $\mathcal{A}^{>}(\Pi)$ in V, otherwise vertices in V are not in $R^-(0)$ in G_{Π}, a contradiction. Consider the graph G_a obtained by taking the subgraph of $G_{\Pi'}$ induced by the vertices in V that are reachable from a, by removing the edge between a and the unique state s with $a \in \mathcal{A}(s)$, and by adding an artificial state s_0 with a as its unique possible action. Let $\mathcal{A}_a$ be the set of actions in G_a.

We can associate a s_0-SSP instance to G_a by considering s as the target state. We can assume w.l.o.g. that Π' is a s_0-proper policy for this problem[8]. Now let $x^{\Pi'}$ be the corresponding flux vector (in principle it is defined only on the actions in G_a but we extend the flux on the other action by setting it to zero). We can interpret $x^{\Pi'}$ as a (non zero) transition cycle of the original problem (the flux is defined on the same set of actions and $x^{\Pi'}(a) = 1$). The vector $x^{\Pi'} \geq 0$ thus satisfies $(J - P)^T x^{\Pi'} = 0$. Now the reduced cost $\bar{c}^{\Pi}(a') = c(a') - c_{\Pi}(I - P_{\Pi})^{-T}(J - P)^T \mathbf{1}_{a'} \leq 0$ for all $a' \in \mathcal{A}_a$ by definition of Π and Π'. Also, as already observed, $\bar{c}^{\Pi}(a) < 0$. Let us analyze $cx^{\Pi'}$. We have $cx^{\Pi'} = \sum_{a'\in\mathcal{A}_a} c(a') \cdot x^{\Pi'}(a') = (\sum_{a'\in\mathcal{A}_a} \bar{c}^{\Pi}(a') \cdot x^{\Pi'}(a')) + c_{\Pi}(I - P_{\Pi})^{-T}(J - P)^T x^{\Pi'}$. Because $(J - P)^T x^{\Pi'} = 0$, we have $c\, x^{\Pi'} = \sum_{a'\in\mathcal{A}_a} \bar{c}^{\Pi}(a')\, x^{\Pi'}(a')$ but this is negative as $x^{\Pi'}(a') > 0$, $\bar{c}^{\Pi}(a') \leq 0$ for all $a' \in \mathcal{A}_a$, and $\bar{c}^{\Pi}(a) < 0$. Therefore $x^{\Pi'}$ is a negative cost transition cycle for our original instance, but this contradicts Assumption 1.

$c\, x^{\Pi'} - c\, x^{\Pi} = c\, (x^{\Pi'} - x^{\Pi}) = (\bar{c}^{\Pi} + c_{\Pi}(I - P_{\Pi})^{-T}(J - P)^T)(x^{\Pi'} - x^{\Pi})$. But by feasibility $(J-P)^T x^{\Pi'} = (J-P)^T x^{\Pi}$ and thus $c\, x^{\Pi'} - c\, x^{\Pi} = \bar{c}^{\Pi}\, (x^{\Pi'} - x^{\Pi}) = \bar{c}^{\Pi}\, x^{\Pi'}$ (as $\bar{c}^{\Pi} = 0$ for all $a \in \Pi$ by definition of the current basis) . This latter term is negative as Π' is using at least one action in $\mathcal{A}^{>}(\Pi)$ and the actions in Π have reduced cost zero.

[8] We can assume without loss of generality that every vertex in G_a is in $R^-(s)$. If not, we change the set V by considering instead the vertices in G_a that do not have a path to s (and we iterate the procedure if V still does not satisfy the required property). Now it is clear that G_a contains at least one state s' and one action $\Pi(s')$. Again not all actions in G_a are actions from Π because otherwise Π would not be proper.