European Journal of Operational Research xxx (xxxx) xxx

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

The Stochastic Shortest Path Problem: A polyhedral combinatorics perspective

Matthieu Guillot[a], Gautier Stauffer[b,*]

[a] Univ. Grenoble Alpes, CNRS, Grenoble INP*, G-SCOP, 38000 Grenoble, France
[b] The Centre of Excellence in Supply Chain (CESIT), KEDGE Business School, Talence, France

ARTICLE INFO

Article history:
Received 6 February 2018
Accepted 29 October 2018
Available online xxx

Keywords:
Markov processes
Stochastic shortest path
Value iteration
Policy iteration
Dijkstra

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 (1991) and by Bertsekas and Yu (2016). 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, i.e., 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.

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 rows and n columns and general term $J(a, s)$, for all $a \in \{1, \ldots, m\}$ and $s \in \{1, \ldots, n\}$, with $J(a, s) = 1$ if and only if action a is available in state s, P is a *row substochastic matrix*[2] with m rows and n columns and general term $P(a, s) := p(s|a)$ (probability of ending in s when taking action a), for all $a \in \{1, \ldots, m\}$, $s \in \{1, \ldots, n\}$, and a cost vector $c \in \mathbb{R}^m$ (see Fig. 1 for an example). The state 0 is called the *target* state and the action 0 is the unique action available in that state. Action 0 leads to state 0 with probability 1. When confusion may arise, we 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 $s \in \{1, \ldots, n\}$ and we assume without loss of generality[3] 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*. We sometimes call the vertices/nodes of G in $\mathcal{S}$ the *state nodes* and the vertices/nodes of G in $\mathcal{A}$ the *action nodes*. A $\mathcal{S}$-*walk* in G is a sequence of vertices $(s_0, a_0, s_1, a_1, \ldots, s_k)$ 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 $0 \leq i \leq k-1$, and $(a_{i-1}, s_i) \in E$ for all $1 \leq i \leq k$. k is called the *length* of the walk. s_k is called the *head* of the walk. We denote by W_k the set of

* Corresponding author at: Institute of Engineering Univ. Grenoble Alpes.
E-mail addresses: matthieu.guillot@g-scop.grenoble-inp.fr (M. Guillot), gautier.stauffer@kedgebs.com (G. Stauffer).

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

[2] Observe that it is usually not a stochastic matrix as state 0 and action 0 are left out.

[3] If not we simply duplicate the actions.

https://doi.org/10.1016/j.ejor.2018.10.052
0377-2217/

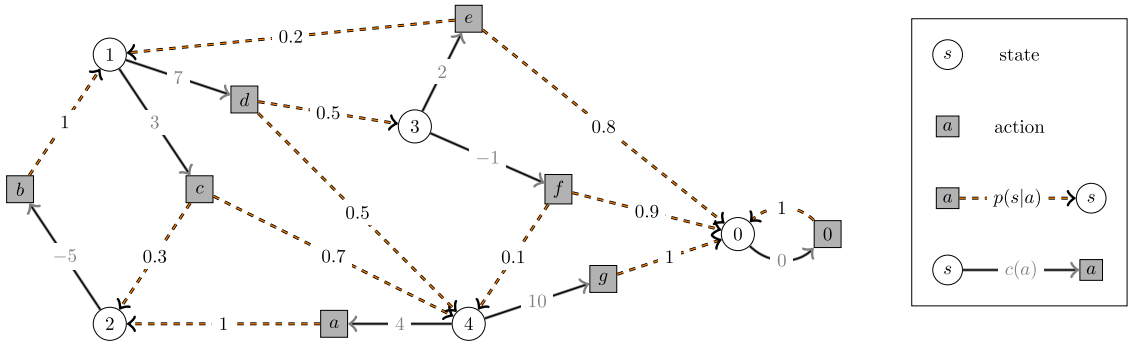

Fig. 1. A graphical representation of a SSP (with target state 0): circles are states, squares are actions, dashed arrows indicate state transitions (probabilities) for a given action, and black edges represent actions available in a given state with corresponding cost.

all possible $\mathcal{S}$-walk of length k and $W := \cup_{k\in\mathbb{N}} W_k$. A *policy* Π is a function $\Pi : (k, w_k) \in \mathbb{N} \times W \mapsto \Pi_{k,w_k} \in \mathcal{M}_=(n, m)$ satisfying $w_k \in W_k$, and $\Pi_{k,w_k}(s, a) > 0 \Rightarrow J(s, a) = 1$ for all $s \in \{1, \dots, n\}$ and $a \in \{1, \dots, m\}$. 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 there exist k and $w_k, w'_k \in W_k$ that share a same head and such that $\Pi_{k,w_k} \neq \Pi_{k,w'_k}$, we say that the policy is *history-dependent* (otherwise it is usually said to be memoryless or *Markovian*). If Π is a constant function, we say that the policy is *stationary*. 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 $n \times m$ matrix.

We let $y_k^\Pi \in \mathbb{R}_+^n$ be the substochastic[4] vector representing the state of the system in period k when following policy Π (from an initial distribution y_0^Π). That is $y_k^\Pi(s)$ is the probability of being in state s, for all $s = 1, \dots, n$ at time k following policy Π. Similarly, we denote by $x_k^\Pi \in \mathbb{R}_+^m$ the substochastic[5] vector representing the probability to perform action a, for all $a = 1, \dots, m$, at time k following policy Π. Given a stationary policy Π and an initial distribution y_0^Π at time 0, by the law of total probability (and because each action is available in exactly one state), we have $x_k^\Pi = \Pi^T \cdot y_k^\Pi$ for all $k \geq 0$. Similarly, 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 $s \in \mathcal{S} \setminus \{0\}$, $J_\Pi(s) := \limsup_{K\to+\infty} \sum_{k=0}^K c^T x_k^\Pi$ with $y_0^\Pi := e_s$, and $J^*(s) := \min\{J_\Pi(s) : \Pi$ deterministic and stationary policy$\}$[6] (e_s is the characteristic vector of $\{s\}$ i.e., the 0/1 vector with $e_s(s') = 1$ iff $s' = s$). Bertsekas and Tsitsiklis (1991) introduced the notion of *proper* stationary policies: a stationary policy Π is said to be *proper* if $\mathbf{1}^T(P^T \cdot \Pi^T)^n \cdot e_s < 1$ for all $s = 1, \dots, n$, that is, after n periods of time, the probability of reaching the target state is positive, from any initial state s. We say that such policies are *BT-proper* (*BT-improper* otherwise) as we will introduce a slight generalization later. Bertsekas and Tsitsiklis (1991) defined a stationary policy Π^* to be *optimal*[7] if $J^*(s) = J_{\Pi^*}(s)$ for all $s \in \mathcal{S} \setminus \{0\}$. They introduced the *Stochastic Shortest Path Problem* as the problem of finding such an optimal stationary policy.

1.1. 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 (Bertsekas, 2005; 2012; Puterman, 2014). 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 (Alterovitz, Simon, & Goldberg, 2008). It has also many applications in operations research, artificial intelligence and economics: from inventory control, reinforcement learning to asset pricing (see for instance Bäuerle & Rieder, 2011; Merton, 1973; Sutton & Barto, 1998; White., 1993). 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 (1962) in the context of pursuit-evasion games and it was later studied thoroughly by Bertsekas and Tsitsiklis (1991).

MDPs were first introduced in the 1950s by Bellman (1957) and Shapley (1953) and they have a long, rich and successful history (see for instance Bertsekas, 2005; Bertsekas, 2012; Puterman, 2014). For most MDPs, it is known that there exists an optimal deterministic and stationary policy (Puterman, 2014). 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 (Bellman, 1957; Howard, 1960). 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 of contraction mappings and Banach fixed-point theorem (Bertsekas, 2005) 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 (Denardo, 1970; Manne, 1960). 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 (Denardo, 1970; d'Epenoux., 1963; Hernández-Lerma & Lasserre, 2002; Hordijk & Kallenberg, 1979; Manne, 1960).

In the context of the SSP, some hypothesis are required for standard methods and proof techniques to apply. Bertsekas and Tsitsiklis (1991) 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 s for which $J_\Pi(s) = +\infty$. In particular they show that one can restrict to de-

[4] It is in general not a purely stochastic vector as state 0 is left out.
[5] It is in general not a purely stochastic vector as action 0 is left out.
[6] lim sup is used here as the limit need not be defined in general.
[7] Note that it is not clear, a priori, whether such a policy exists.

Please cite this article as: M. Guillot and G. Stauffer, The stochastic shortest path problem: A polyhedral combinatorics perspective, European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2018.10.052

terministic policies. Their assumptions naturally discriminate between BT-proper and BT-improper policies. Exploiting further the discrepancy between these policies, (Bertsekas & Yu, 2016) 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, titled Stochastic and Safety Shortest Path problem (Teichteil-Königsbuch., 2012), is a special case of Bertsekas and Yu's framework (it is a bi-objective problem that can be easily modeled 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[8] is a major open problem for MDPs (see for instance Ye, 2011). 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 (Ye, 2005) and then the same author showed that the original policy iteration method proposed by Howard was actually strongly polynomial too Ye (2011) (the analysis was later improved (Hansen, Miltersen, & Zwick, 2013)). The problem is still open for the undiscounted case but Policy Iteration is known to be exponential in that setting (Friedmann, 2009). In contrast, value iteration was proved to be exponential even for the discounted case (Feinberg & Huang, 2014). 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 (1998). 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 (Powell, 2007). 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 developing new exact approaches is thus of great practical interest.

In this paper, we propose an extension of the frameworks of Bertsekas and Tsitsiklis (1991) and Bertsekas and Yu (2016). 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 nonnegative.

[8] A polynomial in the number of states and the number of actions.

Remark

We became aware recently (Quilliot, 2017) that some related results were published in 1990 by Bendali and Quilliot (1990). Similarly to Bertsekas and Tsitsiklis (1991), they extended the shortest path problem to stochastic environments. Their approach was different though: they studied the natural extensions of directed graphs to the stochastic setting (they named the corresponding extensions stochastic networks) and they studied the extensions of arborescences and cycles and their role in an alternative notion of stochastic shortest path. They could prove results similar to Bertsekas and Tsitsiklis (1991) (namely that linear programming could be used to solve the problem and they also described a method similar to Value Iteration). They also proposed an extension of Dijkstra's algorithm. These results were totally unknown to the MDP community until now, probably due to the fact that they were published in French and in a different community (Bendali and Quilliot were apparently equally unaware of the work of Bertsekas & Tsitsiklis (1991)). While there are non empty intersections with our work too (in particular for Dijkstra's algorithm), our results are more general: our framework (strictly) encapsulates both the frameworks of Bertsekas and al. and of Bendali and Quilliot. Besides, our results were proved independently with different techniques.

1.2. Notations and definitions

For a graph G, we denote by $V(G)$ the vertex/node set of G and by $E(G)$ its edge set. 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 $v \in V \setminus S$ 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 say that a graph $G(V, E)$ is *strongly connected* if for all $u, v \in V(G)$, we have $u \in R^+(v)$ and $v \in R^+(u)$. 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 $\mathbb{1}_A(a) = 1$ if and only if $a \in A$. For a vector $x \in \mathbb{R}^d$ and $I \subseteq \{1, \ldots, d\}$ we denote by $x[I]$ the restriction of x to the indices in I and $x(I) := \Sigma_{i \in I} x(i)$. For $s \in \{1, \ldots, n\}$, we denote by e_s the 0/1 vector of $\mathbb{R}^n$ with $e_s(i) = 1$ if and only if $i = s$.

1.3. Main contributions and roadmap

In this paper, we revisit the Stochastic Shortest Path problem, a well-known problem in Markov Decision Processes. We shed some new light on this well-established problem, both structurally and algorithmically. Our approach is to mimic the polyhedral analysis of the deterministic shortest path problem.

On the structural side, we study the polyhedra associated with the natural linear relaxation of the problem. We show that extreme points of the polyhedra are associated with deterministic and stationary policies by generalizing the flow decomposition property (a fundamental result in network flow theory). This allows to: (i) formally prove that we can restrict to such policies for this problem, a fact that was somewhat taken for granted in earlier works; (ii) relax the conditions under which the problem is well-defined and (weakly) polynomial: this is the case now when there is a way to reach the target from any initial state, and there is no 'transition cycle' of negative cost (the extension of negative cost cycles to the stochastic setting); (iii) simplify the analysis of the problem.

Please cite this article as: M. Guillot and G. Stauffer, The stochastic shortest path problem: A polyhedral combinatorics perspective, European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2018.10.052

On the algorithmic side, building upon our polyhedral findings: (i) we prove that the two standard methods for MDPs, i.e., Value Iteration and Policy Iteration, converge in our more general setting; (ii) we also give a new iterative algorithm based on the standard primal-dual algorithm for linear programming. When the costs are nonnegative, this algorithm can be seen as a generalization of Dijkstra's algorithm to the stochastic setting.

We believe that our result closes some important algorithmic and structural gap between the deterministic problem and the stochastic extension. All in all, our new approach allows to generalize, unify and simplify most results on the SSP for finite state/action spaces and we believe that we set the appropriate and natural framework to study the problem in this case.

Besides, our approach has several strengths with respect to the literature: (i) Approaching the problem from a polyhedral combinatorics perspective is new. Polyhedral combinatorics has been a powerful tool in harmonizing and simplifying many fundamental results in combinatorial optimization. We believe that this new perspective on the problem might help address important remaining open questions such as the existence of a strongly polynomial time algorithm; (ii) Our framework properly encapsulates the deterministic shortest path problem, in contrast with prior works; (iii) Our proofs are elementary for people familiar with network flow theory and it should thus provide a new entry point to the problem for people in combinatorial optimization not familiar with Markov Decision Processes. This should help grasping further interest from this community; (iv) Our generalization allows to capture many important subproblems that were not fitting in the previous frameworks, such as the so-called MAXPROB problem, where the goal is to reach a target with maximum probability: this is a core problem in optimal control, artificial intelligence and game theory.

We now try to give an overview of the different propositions, lemmas and theorems that follow. This should help the reader familiar with the deterministic shortest path problem and its relation with network flow theory to navigate smoothly through the next sections.

In Section 2, we introduce our new framework for the stochastic shortest path problem. We show in particular (Lemma 1) that our framework properly encapsulates the one proposed by Bertsekas and Tsitsiklis (1991). In the deterministic setting the standard assumptions for the shortest (s, t)-path problem are that (i) there exists a path from any node to t and (ii) there is no negative cost cycle. Assumption 5 generalizes these two assumptions to the stochastic setting in the most natural way and we prove that the corresponding assumptions are easily checked (Lemmas 2–4). We also introduce the natural linear programming relaxation of the problem, the so-called network flux relaxation (see (P_s)). This is again the natural generalization of the network flow formulation for the deterministic version.

In Section 3, we prove that the network flux relaxation is actually a formulation by showing that the extreme points are associated with *proper*,[9] deterministic and stationary policies (Corollary 13). This result builds upon an extension of a fundamental theorem for network flows: *the flow decomposition theorem*. The idea in the deterministic case is to decompose a flow in paths and cycles. In the stochastic setting this translates into a decomposition in terms of proper, deterministic and stationary policies and *transition cycles*. The proof of the corresponding theorem (Theorem 10) builds upon different basic properties of *flux vectors* (that is, solutions to the network flux relaxation), namely, that if a flux 'goes through' a node, then this node has to be connected to the sink (Proposition 6), and the fact that one can easily associate a flux vector to a proper, deterministic and stationary policy (Proposition 7). These properties are trivial when instantiated in the deterministic setting but a bit more technical in the stochastic case. As in the deterministic case, the idea behind the flow decomposition theorem is to first identify "paths" (Lemma 11) and then decompose "cycles" (Lemma 12). Lemma 9 shows that each step in the decomposition can be performed efficiently. Finally Lemma 14 generalizes Bellman optimality conditions to the stochastic setting (that is, if the optimal policy *can* visit a certain node, the policy should be optimal from this node too).

Bellman optimality conditions are then exploited in Section 4 to derive several iterative algorithms. We first prove that Value Iteration (the generalization of Bellman–Ford algorithm), converges in value to the optimal solution (Theorem 15) and we show that we can in fact extract a series of proper, deterministic and stationary policies that converge to the optimal policy (Theorem 16). We then prove that we can exploit the linear programming formulation to derive simplex-type of algorithms: any standard (single pivot) simplex method will find an optimal policy in a finite number of steps as the linear program is non degenerate (Theorem 17), and Howard's Policy Iteration method (a multi-pivot simplex algorithm) also converges in a finite number of steps (Theorem 19). The latter result exploits the fact that the objective function is nonincreasing in each iteration (Proposition 18). Finally Theorem 20 exploits the standard primal-dual algorithm for linear programming to provide a natural extension of Dijkstra's algorithm.

2. Our new framework

We start with a simple observation whose proof can be found in A.1.

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 $s \in \{1, \dots, n\}$, 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 $\lim_{k\to+\infty} y_k^\Pi = 0$ and thus the policy leads to the target state 0 with probability 1 from state s. A s-proper policy is thus a policy that converges to the target with probability one and whose expected number of visit in each action is finite. The expected cost of such a policy is thus the well-defined value $c^T \sum_{k\geq 0} x_k^\Pi$. 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 it is called *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 $\{1, \dots, n\}$ 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 $k \geq 0$, $\sum_{a\in\mathcal{A}(s)} x_k^\Pi(a) = y_k^\Pi(s)$ (remember that each action is available in exactly one state). We also have $y_{k+1}^\Pi(s) = \sum_{a\in\mathcal{A}} p(s|a) x_k^\Pi(a)$ for all $s \in \{1, \dots, n\}$. 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

[9] The new definition of proper is given in the beginning of the next section.

Please cite this article as: M. Guillot and G. Stauffer, The stochastic shortest path problem: A polyhedral combinatorics perspective, European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2018.10.052

linear program is a relaxation of the s-SSP problem.[10]

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

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 as it does not contain the row associated with the sink node) and the corresponding LP is the standard network flow relaxation of the deterministic shortest path problem (again up to a row as we remove the (redundant) flow conservation constraint for the sink node). 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. 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 MDP theory "Under what conditions is it optimal to restrict to deterministic and stationary policies?" Puterman (2014).

We can assume without loss of generality that there exists a path between all state node s' and 0 in the support graph G. Indeed, if there is a state node s' with no path to 0 in G, then no s-proper policy will pass through s' at any point in time (because then the probability of reaching the target state, starting from s', is zero, contradicting $\lim_{k\to+\infty} y_k^\Pi = 0$); we could thus remove s' and the actions leading to s' and iterate. Under this assumption, there is always a s-proper policy. Indeed the randomized and stationary policy Π that chooses an action uniformly at random among $\mathcal{A}(s')$, in each state $s' = 1, \ldots, n$, will work: in this case, for each state s', there is in fact a non zero probability of choosing one of the paths from s' 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 s' and 0 in the support graph G. Then the policy that consists, for each state $s' \in \mathcal{S} \setminus \{0\}$, in choosing uniformly at random an action in $\mathcal{A}(s')$ 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 $O(|\mathcal{S}| \cdot (|\mathcal{S}| + |\mathcal{A}| + |E|))$ whether a s-SSP instance with support graph $G = (\mathcal{S}, \mathcal{A}, E)$ 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 5. We consider s-SSP/SSP instances where:

- there exists a path between all state node s' 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, $J^*(s')$ is not real-valued for some state s'.[11] The main extension, with respect to Bertsekas and Yu, is that we allow for non-stationary proper policies in the first place.

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 5, we can restrict to optimal proper, deterministic and stationary policies.

We start with a few definitions. Let $G = (\mathcal{S}, \mathcal{A}, E)$ be the support graph of our s-SSP instance and let $x \in \mathbb{R}^m$. We define G_x to be the subgraph of G induced by the vertices in $\mathcal{A}_x \cup N_G^+(\mathcal{A}_x) \cup N_G^-(\mathcal{A}_x)$ where $\mathcal{A}_x := \{a \in \{1, \ldots, m\} \text{ with } x(a) > 0\}$. G_x is called the *support graph of x in G*. Again we call *state nodes* the vertices/nodes of G_x that are in $\mathcal{S}$ and *action nodes* the vertices/nodes of G_x that are in $\mathcal{A}$. We denote by E_x the set of edges of G_x. A transition cycle x is *simple* if for all state nodes s' in $V(G_x)$, there exists exactly one edge of $N_G^+(s')$ in E_x, i.e., $|N_{G_x}^+(s')| = 1$, and G_x is strongly connected.

The main theorem of this section is an extension of the *flow decomposition theorem*, which is a fundamental result in network flow theory (see Ahuja, Magnanti, & Orlin, 1993). It asserts that any network flux is a convex combination of network flux 'associated with' proper policies plus a conic combination of simple transition cycles (see Theorem 10). Before we can prove this theorem, we need a couple of useful propositions. The proof of the first proposition builds upon simple flow conservation arguments, we defer the proof to A.2.

Proposition 6. *Let $x \in \mathbb{R}^m$ be a feasible solution of (P_s). There exists a path between all states reachable from s in G_x and $0_\mathcal{S}$. In other word, for all $s' \in R_{G_x}^+(s)$, we have $s' \in R_{G_x}^-(0_\mathcal{S})$.*

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_Π^s be the subgraph of G_Π induced by the vertices in $R_{G_\Pi}^+(s)$. G_Π^s 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 when starting from s). Because Π is proper, $0_\mathcal{S}$ is reachable from each state s' in G_Π^s. Let us denote by $\mathcal{S}'$ the state vertices in G_Π^s 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 columns in $\mathcal{S}'$ and the rows in $\Pi(\mathcal{S}')$ (since Π is deterministic, $P_{\mathcal{S}'}$ is a $|\mathcal{S}'| \times |\mathcal{S}'|$ matrix). $P_{\mathcal{S}'}^T$ can be interpreted as the transition matrix associated with $\mathcal{S}'$ when following policy Π (we do not leave $\mathcal{S}'$): $P_{\mathcal{S}'}(\Pi(s'), s'')$ gives the probability of ending in state s'' (in one iteration) when starting in state

[10] 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 Bertsekas & Yu, 2016 for instance). However while usually, 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 Hansen (2012).

[11] One can prove using Lemma 12 and basic geometry that when there exists a negative cost transition cycle, there exists also a *simple* transition cycle of negative cost (see the definition in the second paragraph of Section 3): all state nodes s' on this cycle will have $J_\Pi(s') = -\infty$, where Π is the deterministic and stationary policy that consists in choosing the unique action $a \in \mathcal{A}(s')$ with $x(a) > 0$ for each state node s' on the cycle and any action for the state nodes that are not on the cycle.

Please cite this article as: M. Guillot and G. Stauffer, The stochastic shortest path problem: A polyhedral combinatorics perspective, European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2018.10.052

s' and using $\Pi(s')$. Hence, if we assume that the rows of $P_{S'}$ are ordered according to S', then $P_{S'}^T e'_i$ defines the state of the system after one iteration of policy Π if we start in state $i \in \mathcal{S}'$ (e'_i is the restriction of e_i to to the indices in $\mathcal{S}'$). Now as already observed, $0_{\mathcal{S}}$ is reachable from any node in $\mathcal{S}'$ and it thus follows that $(P_{S'}^T)^k e'_i$, the state of the system after k steps, tends to zero as k tends to infinity (remember that $0_{\mathcal{S}}$ is left out). Because this is true for any $i \in \mathcal{S}'$, we have $\lim_{k\to+\infty}(P_{S'}^T)^k = 0$ and thus $(I_{S'} - P_{S'})$ is invertible by Lemma 21. Now observe that $(I_{S'} - P_{S'})^T x^\Pi[\Pi(S')] = e'_s$ for $x^\Pi := \sum_{k=0}^{+\infty} x_k^\Pi$, with $y_0^\Pi := e_s$. Indeed $x^\Pi(a) = 0$ for all $a \notin \Pi(S')$ and thus $(I_{S'} - P_{S'})^T x^\Pi[\Pi(S')] = e'_s$ corresponds to the constraints of (P_s) associated with the rows in $\mathcal{S}'$. We thus have the following result.

Proposition 7. *Given a proper, deterministic and stationary policy Π, the flux vector x^Π associated with Π and defined by $x^\Pi := \sum_{k=0}^{+\infty} x_k^\Pi$, with $y_0^\Pi := e_s$ satisfies $x^\Pi[\Pi(S')] = (I_{S'} - P_{S'})^{-T} e'_s$ and $x^\Pi(a) = 0$ for all $a \notin \Pi(S')$, with $\mathcal{S}'$, $\Pi(\mathcal{S}')$, $I_{S'}$, $P_{S'}$ and e'_s defined as above.*

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

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

Before proving Theorem 10, we need a final lemma.

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

Proof. We know that, $0_{\mathcal{S}} \in R^+(s')$ for all s', is enough to ensure that there is a proper policy by Lemma 3. Now if there exists a state vertex s' in G with $|\mathcal{A}(s')| > 1$, we can delete from G an action in $\mathcal{A}(s')$ that does not remove $0_{\mathcal{S}}$ from $R^+(s')$. Such an action exists as it is enough to keep an action $a \in A(s')$ with minimum distance to 0 (in terms of arc) to ensure that $0_{\mathcal{S}}$ is still in $R^+(s')$ after deletion (by minimality of the distance to 0, such an action has a directed path to $0_{\mathcal{S}}$ that does not go through s'). If $|\mathcal{A}(s')| = 1$ for all s' then the only possible policy is proper (from Lemma 3), deterministic and stationary. We can implement such a procedure in time $O(|\mathcal{S}| + |\mathcal{A}| + |E|)$ by computing $N_k^-(0)$ for all $k \leq |\mathcal{S}| + |\mathcal{A}|$ and a $0_{\mathcal{S}}$-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 $k, k' \in \mathbb{N}$ with $1 \leq k$, $k + k' \leq m$, $x_1, \ldots, x_k \in \mathbb{R}^m$, $x'_1, \ldots, x'_{k'} \in \mathbb{R}^m$, $\lambda_1, \ldots, \lambda_k \in [0, 1]$, and $\lambda'_1, \ldots, \lambda'_{k'} \geq 0$ such that $x_1, \ldots, x_k$ are feasible solutions of (P_s), $x'_1, \ldots, x'_{k'}$ are simple transition cycles, $\sum_{j=1}^k \lambda_j = 1$ and $x = \sum_{j=1}^k \lambda_j x_j + \sum_{j'=1}^{k'} \lambda'_{j'} x'_{j'}$. 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 will start with a slightly simpler version. □

Lemma 11. *Let $x \in \mathbb{R}^m$ be a feasible solution of (P_s). In strongly polynomial time, one can find $k \in \mathbb{N}$, $x_1, \ldots, x_k, x_c \in \mathbb{R}^m$,and $\lambda_1, \ldots, \lambda_k \in [0, 1]$ such that $1 \leq k \leq m - |\mathcal{A}_{x_c}|$, $x_1, \ldots, x_k$ are feasible solutions of (P_s), x_c is a transition cycle, $\sum_{j=1}^k \lambda_j = 1$ and $x = \sum_{j=1}^k \lambda_j x_j + x_c$. 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 $|\mathcal{A}_x|$). Because x is a feasible solution of (P_s), we know by Proposition 6 that there exists a path between all states reachable from s in G_x and $0_{\mathcal{S}}$. 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 Proposition 7. Let $1 \geq \lambda \geq 0$ be the maximum value such that $x' := x - \lambda x^\Pi \geq 0$. By Proposition 8 we have that G_{x^Π} is a subgraph of G_x and thus $\lambda > 0$ (as $x > 0$ on $\mathcal{A}_x$). 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 (note that $|\mathcal{A}_{x'}| < |\mathcal{A}_x|$ as $\mathcal{A}_{x'}$ must miss at least one action of Π leading to $0_{\mathcal{S}}$ with non zero probability). If $\lambda < 1$, by maximality of λ, there is an arc $a \in \mathcal{A}_x$ such that $x(a) > 0$ and $x'(a) = 0$. Hence $\mathcal{A}_{x'} \subset \mathcal{A}_x$ and $\frac{1}{1-\lambda} x'$ 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 a contradiction. Clearly, we can make the proof algorithmic and because $A_{x'} \subset A_x$ at each iteration, the algorithm will terminate with a set of k solutions $x_1, \ldots, x_k$ to (P_s) and a vector x_c satisfying the theorem in at most $|\mathcal{A}_x| - |\mathcal{A}_{x_c}| \leq m - |\mathcal{A}_{x_c}|$ steps. □

The following lemma builds upon similar ideas.

Lemma 12. *Let $x' \neq 0 \in \mathbb{R}^m$ be a transition cycle. In strongly polynomial time, one can find $k' \in \mathbb{N}$, $x_1, \ldots, x_{k'}, \in \mathbb{R}^m$,and $\lambda'_1, \ldots, \lambda'_{k'} \geq 0$ such that $1 \leq k' \leq |\mathcal{A}_{x'}|$, $x'_1, \ldots, x'_{k'}$ are simple transition cycles and $x' = \sum_{j=1}^k \lambda'_{j'} x'_{j'}$.*

Proof. We prove first that such a decomposition exists for any transition cycle $x' \neq 0$. Let x' be a smallest counter-example (in terms of $|\mathcal{A}_{x'}|$). We focus on the support graph $G_{x'}$. By minimality of the counter-example, we can assume that $G_{x'}$ is connected. Now $G_{x'}$ has to be strongly connected otherwise it would contradict flow conservation constraints (using similar argument as in Proposition 6). Observe also that $0_{\mathcal{S}}$ is not in $V(G_x)$. Let us consider any action a in $\mathcal{A}_{x'}$ and let us call e the edge between a and the unique node s with $a \in \mathcal{A}(s)$. We can consider the graph G_a obtained by taking the subgraph of $G_{x'} \setminus e$ induced by the vertices that are reachable from a (in $G_{x'} \setminus e$), by 'splitting' action a. Let s be the unique state where a is available. We add two artificial node states s_0, t_0 and an artificial action a_0 that leads to t_0 with probability 1, such that a is removed from the set of actions available in s and a becomes the only action available in s_0. Let G'_a be the corresponding graph. We can consider an instance of s_0-SSP with target state t_0 in G'_a. Now x' can easily be converted into a feasible network flux $\bar{x}$ for the corresponding problem by simply setting $\bar{x}(a') = \frac{x'(a')}{x'(a)}$ for all $a' \neq a_0$ and $\bar{x}(a_0) = 1$. We can then apply Lemma 11 to $\bar{x}$ to generate $x_1, \ldots, x_k$, $\lambda_1, \ldots, \lambda_k$ and x_c obeying the corresponding lemma. Now we can convert $x_1, \ldots, x_k$ into simple transition cycles of our original instance by setting, for all $i = 1, \ldots, k$ and for all $a' \neq a_0$, $x'_i(a') = x_i(a')$. It follows that $x' = x'(a) \cdot (\sum_{i=1}^k \lambda_i x'_i + x_c)$. Remember that $k \geq 1$, so x'_1 exists. Now for $\mu = \min_{a'}\{\frac{x'(a')}{x'_1(a')}\}$, $x'' = x' - \mu x'_1$ is still a transition cycle, but $|\mathcal{A}_{x''}| < |\mathcal{A}_{x'}|$ by the choice of μ, so by minimality of the counter-example, x'' can be decomposed into simple transition cycles and so $x' = x'' + \mu x'_1$ too, a contradiction. We can again make the proof algorithmic and so $k' \leq |A_{x'}|$. □

Theorem 10 is a direct corollary of Lemmas 11 and 12: we apply Lemma 12 to the transition cycle x_c returned by Lemma 11.

We can now exploit Theorem 10 to prove that under our assumptions, we can restrict to deterministic and stationary policies.

Please cite this article as: M. Guillot and G. Stauffer, The stochastic shortest path problem: A polyhedral combinatorics perspective, European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2018.10.052

Corollary 13. *Under Assumption 5, the s-SSP admits an optimal proper, deterministic and stationary policy.*

Proof. We know from linear programming that when a LP has a finite optimum, we can find an optimal solution in an extreme point. For (P_s), existence of of finite optimum is guaranteed by Assumption 5: the first conditions implies the existence of a solution by Lemma 3 and the second conditions ensure that the value is bounded. 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 precedes a result which is standard for the deterministic shortest path problem: *Bellman optimality conditions*.

Lemma 14. *Let Π be an optimal proper, deterministic and stationary solution to the s-SSP (under Assumption 5). Let G^s_Π be the support graph of Π. For all state vertex s' in G^s_Π, Π is optimal for s'-SSP.*

Proof. Observe first that s'-SSP satisfies Assumption 5. Now suppose Π is not optimal for s'-SSP. We know from Corollary 13 that s'-SSP admits an optimal proper, deterministic and stationary policy $\Pi_{s'}$. Now the (history-dependent and non stationary) policy Π' that consists in applying policy Π to problem s-SSP, up to when state s' is reached (if it ever is) and then applying policy $\Pi_{s'}$ is a proper policy. The value of this policy is (strictly) better than the value of Π as there exist realizations where s' 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 14) also tells us that, under Assumption 5, 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 $i = 1, \dots, n$ 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 14, an optimal policy Π^* for SSP is optimal for s'-SSP for all $s' = 1, \dots, n$ (as all s' 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 $x^\Pi := \sum_{k=0}^{+\infty} x^\Pi_k$ with $y^\Pi_0 := \frac{1}{n}\mathbf{1}$ and it satisfies $x^\Pi = (I - P_\Pi)^{-T}\frac{1}{n}\mathbf{1}$ (see Proposition 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):

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

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 5. 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 $s = 1, \dots, n$, we define $V^*(s)$ to be the optimal value of (P_s) (again under Assumption 5), i.e., $V^*(s) := \min_{\Pi \in \mathcal{P}} c^T x^\Pi$ with $y^\Pi_0 = e_s$. This is referred to as the *value* of state s. We have in particular $V^*(s) = \min_{\Pi \in \mathcal{P}} \lim_{K \to +\infty} \sum_{k=0}^K c^T x^\Pi_k$ by definition of x^Π_k. 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 $M(s)$ for each state $s = 1, \dots, n$ that lead to state 0 with probability one, with $M(s)$ "big enough". We call aux-SSP this auxiliary problem (we slightly abuse notation and we still denote by c the corresponding cost function). 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 $V_K(s) := \min_{\Pi \in \mathcal{P}^K} \sum_{k=0}^K c^T x^\Pi_k$ is well-defined for each $K \geq 1$. In fact we can prove by induction that it follows the dynamic programming formula: $V_k(s) = \min\{V_{k-1}(s), \min_{a \in \mathcal{A}(s)} c(a) + \sum_{s'} p(s'|a) V_{k-1}(s')\}$ for all $k \geq 2$ and $V_1(s) = M(s)$ for all $s = 1, \dots, n$ (an optimal, deterministic non-stationary policy Π^*_K can be recovered easily too): $V_k(s)$ is indeed the optimal value starting from s 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.

Theorem 15. *For all $s = 1, \dots, n$, if $M(s) \geq V^*(s)$, then we have $V^*(s) = \lim_{K \to +\infty} V_K(s)$.*

Proof. We will prove that $\min_{\Pi \in \mathcal{P}} \lim_{K \to +\infty} \sum_{k=0}^K c^T x^\Pi_k = \lim_{K \to +\infty} \min_{\Pi \in \mathcal{P}^K} \sum_{k=0}^K c^T x^\Pi_k$ with $y^\Pi_0 := e_s$, for all $s = 1, \dots, n$, by proving both inequalities.

$\boxed{\leq}$ Let Π^*_K be an optimal solution to $\min_{\Pi \in \mathcal{P}^K} \sum_{k=0}^K c^T x^\Pi_k$ 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 $V_K(s) = c^T \sum_{k=0}^K x^{\Pi^*_K}_k \geq \min_{\Pi \in \mathcal{P}_{aux}} \lim_{K \to +\infty} \sum_{k=0}^K c^T x^\Pi_k$ (observe that this minimum is well defined since we are still satisfying Assumptions 5 in aux-SSP). By construction $\{V_K(s), K \geq 1\}$ is nonincreasing, hence because it is bounded from below, it converges and $\lim_{K \to +\infty} V_K(s)$ is well-defined. Taking the limit we get $\lim_{K \to +\infty} c^T \sum_{k=0}^K x^{\Pi^*_K}_k \geq \min_{\Pi \in \mathcal{P}_{aux}} \lim_{K \to +\infty} \sum_{k=0}^K c^T x^\Pi_k$. But $\min_{\Pi \in \mathcal{P}_{aux}} \lim_{K \to +\infty} \sum_{k=0}^K c^T x^\Pi_k \geq \min_{\Pi \in \mathcal{P}} \lim_{K \to +\infty} \sum_{k=0}^K c^T x^\Pi_k$ if $M(s)$ is chosen so that auxiliary actions can be assumed not to be used in an optimal policy Π^* for $\mathcal{P}_{aux}$. This is the case for $M(s) \geq V^*(s)$ as we could consider the (non stationary) policy that applies policy Π^* up to the first time we want to use an artificial action and then apply an optimal policy Π^{**} for $\mathcal{P}$: the corresponding policy has a value no greater than the former.

$\boxed{\geq}$ Let Π^* be an optimal proper deterministic and stationary solution to $\min_{\Pi \in \mathcal{P}} \lim_{K \to +\infty} \sum_{k=0}^K c^T x^\Pi_k$ (Π^* exists in our setting by Corollary 13). 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

Please cite this article as: M. Guillot and G. Stauffer, The stochastic shortest path problem: A polyhedral combinatorics perspective, European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2018.10.052

of $\mathcal{P}^K$ obtained from using Π^* in periods $0,\dots,K-1$ 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 s-proper. □

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

The algorithm that consists in computing V_k iteratively 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 (2016).

We now explain how to recover an optimal proper, deterministic and stationary policy, if we were lucky enough to get the optimal vector V^* after some iterations (if we build a feasible policy Π_k at each iteration, it may happen that we discover that Π is optimal by computing V^{Π} and observing that it satisfies the Bellman equations). 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 $V^*(s)$ and by Lemma 14, we know that V^* satisfies $V^*(s) = \min_{a\in\mathcal{A}(s)} c(a) + \sum_{s'} p(s'|a)V^*(s')$ for all $s = 1,\dots,n$. Also extending Corollary 13 to SSP, we know that there exists an optimal proper deterministic and stationary policy Π^* with $V^*(s) = c(\Pi^*(s)) + \sum_{s'} p(s'|\Pi^*(s))V^*(s')$ for all $s = 1,\dots,n$. 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 satisfy $x^*(a) > 0$ only if $V^*(s) = c(a) + \sum_{s'} p(s'|a)V^*(s')$. 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 $G^* = (\mathcal{S}^*, \mathcal{A}^*, E^*)$ of this instance. We know from Lemma 9 that we can thus find a proper, deterministic and stationary policy Π in time $O(|\mathcal{S}^*| + |\mathcal{A}^*| + |E^*|)$. The pair (x^{Π}, y^*) satisfies the complementary slackness conditions and thus Π is optimal.

Unfortunately, we might never reach the precise value of V^* when iterating VI. However, we can build a proper deterministic and stationary policy Π_k at each step k of Value Iteration by considering an approximate version of the complementary slackness theorem. For all $k \geq 0$, we define $y_k := V_k$, and, for each action a, $\epsilon_a^k := V_k(s^{-1}(a)) - (c(a) + \sum_{s'} p(s'|a)V_k(s'))$. For $\epsilon \geq 0$, we define $\mathcal{A}_\epsilon^k$ the set of actions $a \in \mathcal{A}$ such that $\epsilon_a^k \geq -\epsilon$ and we denote by SSP_ϵ^k the restriction of our SSP instance to the actions in $\mathcal{A}_\epsilon^k$. Let us denote by $\epsilon_k \geq 0$ the minimum value $\epsilon \geq 0$ such that SSP_ϵ^k admits a proper, deterministic and stationary policy Π^k. Observe that $\epsilon_k \in \{-\epsilon_a^k, a \in \mathcal{A}\}$. We can thus compute ϵ_k and Π_k in strongly polytime using Lemmas 4 and 9. We will now prove that V^{Π_k} converges to V^* as k tends to infinity ($V^{\Pi_k}(s)$ is the value associated with Π_k when starting from s).

Let us first notice that $\epsilon_k \leq \max_s\{V_k(s) - V^*(s)\}$ (remember $V_k(s) \geq V^*(s)$). Indeed for $\epsilon = \max_s\{V_k(s) - V^*(s)\}$, we have $V^*(s) \leq V_k(s) \leq V^*(s) + \epsilon$ for all s and it follows that for any s and for any optimal policy Π^*, we have $c(\Pi^*(s)) + \sum_{s'} p(s'|\Pi^*(s))V_k(s') \leq c(\Pi^*(s)) + \sum_{s'} p(s'|\Pi^*(s))(V^*(s') + \epsilon) \leq c(\Pi^*(s)) + \sum_{s'} p(s'|\Pi^*(s))V^*(s') + \epsilon = V^*(s) + \epsilon$. It follows that $V_k(s) - (c(\Pi^*(s)) + \sum_{s'} p(s'|\Pi^*(s))V_k(s')) \geq V_k(s) - V^*(s) - \epsilon \geq -\epsilon$ and thus $\Pi^*(s) \in \mathcal{A}_\epsilon^k$. Hence Π^* is a proper deterministic and stationary policy of SSP_ϵ^k and the result follows. It implies in particular that ϵ_k tends to zero as k tends to infinity by Theorem 15.

Let us consider the pair (x^{Π_k}, y_k). x^{Π_k} is a solution of (P) but y_k might not be a feasible solution to (D) so it is not a primal/dual pair of solutions. However it almost satisfies the complementary slackness conditions. In particular we have $\sum_{a\in\mathcal{A}} c(a)x^{\Pi_k}(a) = \sum_{a\in\Pi_k} c(a)x^{\Pi_k}(a) \leq \sum_{a\in\Pi_k}(y_k(J-P)^T\mathbf{1}_a + \epsilon_k)$ $x^{\Pi_k}(a) = y_k(J-P)^T x^{\Pi_k} + \sum_{a\in\Pi_k} \epsilon_k x^{\Pi_k}(a) = y_k\mathbf{1} + \epsilon_k \mathbf{1}^T x^{\Pi_k}$. It follows that, as k tends to infinity, $\sum_{a\in\mathcal{A}} c(a)x^{\Pi_k}(a)$ tends to the optimal value of (P). Indeed y_k tends to V^* by Theorem 15 (and $V^*\mathbf{1}$ is the optimal value of (D) and (P)), ϵ_k tends to zero by the discussion above, and $\mathbf{1}^T x$ is bounded for proper policies. Therefore x_k^{Π} tends to be an optimal solution of P, Π_k tends to be an optimal policy, and V^{Π_k} tends to V^*. We sum up the result in the following theorem.

Theorem 16. *In each iteration k of Value Iteration, one can compute in strongly polynomial time a proper, deterministic and stationary policy Π_k such that V^{Π_k} tends to V^* as k tends to infinity.*

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 (Melekopoglou & Condon, 1994).

Theorem 17. *Under Assumption 5, simple policy iteration converges in a finite number of steps.*

In contrast with simple policy iteration, Howard's original policy iteration method (Howard, 1960) changes the actions of a (basic) policy in each state s for which there is an action in $\mathcal{A}(s)$ with negative *reduced cost*. We will prove now that this method converges under Assumption 5, by proving 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 $\tilde{c}^{\Pi} := c - c_{\Pi}(I - P_{\Pi})^{-T}(J - P)^T$ 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 $\tilde{c}^{\Pi}(a) < 0$. We know from linear programming that if $\tilde{c}^{\Pi}(a) \geq 0$ 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.

Proposition 18. *Π' is proper and $c \cdot x^{\Pi'} < c \cdot x^{\Pi}$*

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 $R^-_{G_{\Pi'}}(0)$. It implies that there is a set of vertices V in $V(G_{\Pi'})$ such that $0_S, 0_{\mathcal{A}} \notin V$ and $\delta^+(V) = \emptyset$. We can choose for instance $V = V(G_{\Pi'}) \setminus R^-_{G_{\Pi'}}(0)$. Now we choose V minimal with this property. There exists an action a of $\mathcal{A}^<(\Pi)$ in V, otherwise vertices in V are not in $R^-_{G_{\Pi}}(0)$, a contradiction with Π being proper. Consider the graph G_a obtained

Please cite this article as: M. Guillot and G. Stauffer, The stochastic shortest path problem: A polyhedral combinatorics perspective, European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2018.10.052

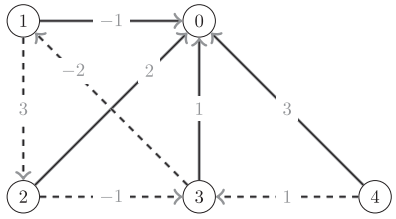

Fig. 2. A deterministic shortest path (with target state 0): the dark "actions" represent the current policy, and the dashed "actions" have non positive reduced cost; changing all actions with non positive reduced cost yield a new policy which is not proper.

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. Note that by minimality of V, every vertex in G_a is in $R^-_{\Pi'}(s)$. Indeed if not we can change the set V by considering instead the vertices in G_a that do not have a path to s.

We can associate a s_0-SSP instance to G_a by considering s as the target state. Π' is a s_0-proper policy for this problem. 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 actions 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 $\bar{c}^{\Pi}(a') = c(a') - c_\Pi (I - P_\Pi)^{-T}(J-P)^T \mathbf{1}_{a'}$ satisfies $\bar{c}^{\Pi}(a') \leq 0$ for all $a' \in \mathcal{A}_a$, by definition of Π and Π' (actions in Π have reduced cost 0 and actions in Π' that are not action from Π have a negative reduced cost). 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 5.

Now that we know that Π' is proper, we can define $x^{\Pi'}$ to be the network flux associated with Π'. We have $c\ x^{\Pi'} - c\ x^{\Pi} = c\ (x^{\Pi'} - x^{\Pi}) = (\bar{c}^{\Pi} + c_\Pi (I - P_\Pi)^{-T}(J-P)^T)(x^{\Pi'} - x^{\Pi})$. But by feasibility of Π' and Π, we have $(J-P)^T x^{\Pi'} = (J-P)^T x^{\Pi} = \mathbf{1}$ and thus $c\ x^{\Pi'} - c\ x^{\Pi} = \bar{c}^{\Pi}\ (x^{\Pi'} - x^{\Pi}) = \bar{c}^{\Pi}\ x^{\Pi'}$ (as $\bar{c}^{\Pi}(a) = 0$ 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. □

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

Theorem 19. *Under Assumption 5, 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 Proposition 18 fails (see for instance Fig. 2). 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 Fearnley (2010).

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 (Edmonds, 1965) is probably the most celebrated example. It is well-known that for the deterministic shortest path problem, when the costs are nonnegative, the primal-dual approach corresponds to Dijkstra's algorithm (Papadimitriou & Steiglitz, 1982). We extend this approach to the SSP setting. Let us first recall the linear formulation of the problem and its dual:

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

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 a finite optimum by Assumption 5). 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).

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

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 (Papadimitriou & Steiglitz, 1982)) 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 $\bar{\mathcal{A}} \cup \{m+1, \ldots, m+n\}$, where actions $m+k$, for all $k = 1, \ldots, n$ 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 $m+1, \ldots, m+n$ 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 (Mausam & Kolobov, 2012). Little is known about the complexity of 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 (Melekopoglou & Condon, 1994): 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 nonnegative costs instances.

Theorem 20. *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 nonnegative to apply the primal-dual approach. In fact, one can use the standard trick of adding an artificial constraint

Please cite this article as: M. Guillot and G. Stauffer, The stochastic shortest path problem: A polyhedral combinatorics perspective, European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2018.10.052

$\Sigma_a x_a \leq M$ to the problem, with M "big" to find an initial dual solution and iterate the algorithm (Papadimitriou & Steiglitz, 1982). 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.

One might consider variants of the primal-dual algorithm where the updates of the dual solution do not follow the generic mechanism that guarantees finiteness for general LPs, but instead the updates are 'ad-hoc' and exploit the structure of the problem. It might still be possible to prove finiteness of the algorithm in such cases. For instance the so-called auction algorithm (Bertsekas, 1991) introduced by Bertsekas to solve the (deterministic) shortest path problem can be seen as such an ad-hoc implementation of the primal-dual algorithm. The original version is pseudo-polynomial but it could be turned into a strongly polynomial time algorithm (Bertsekas, Pallottino, & Scutellá, 1995). This might be an alternative route toward a strongly polynomial time algorithm for the stochastic shortest path problem.

5. Conclusion and perspectives

In this paper, we have introduced a new unifying framework for the stochastic shortest path problem. We have shown that the classic *flow decomposition theorem* extends naturally from network flows to network flux and we have exploited this result to prove that, in this setting, we can restrict to deterministic and stationary policies and that the standard iterative algorithms for Markov Decision Process, i.e., Value Iteration and Policy Iteration, converge. We have also introduced a new promising algorithm that can be seen as a generalization of Dijkstra's algorithm for the deterministic shortest path problem. Our goal is now to implement fast versions of these algorithms and to compare their practical performances on various real-world instances. While the implementation of Value Iteration and Policy Iteration does not seem to provide major numerical challenges (we are still testing the corresponding implementations), our first implementation of the generalization of Dijkstra's algorithm suffers from numerical instability. We are gaining expertise in this area by exchanging with experts of stabilization techniques. We should soon have a stable version of this algorithm implemented. Nevertheless a careful numerical evaluation of the different methods on significative instances is beyond the scope of the current paper. We leave the corresponding project for future research.

This paper leaves several fundamental questions unsolved. In particular, we do not provide new insight on the challenging question of whether the stochastic shortest path problem (or total reward undiscounted MDPs) can be solved in strongly polynomial time. We conclude though with a few (possibly) simpler questions that, we believe, deserve some attention on their own and that might help addressing the former: Is it possible to solve MAXPROB in strongly polynomial time? Can we bound the number of iterations of our variant of Dijkstra's algorithm by a polynomial? by a polynomial in n and m? Can the stochastic shortest path problem be solved in strongly polynomial time when the costs are nonnegative?

A.1. 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 21, it implies that $I - \Pi^T \cdot P^T$ is invertible and that $\lim_{K\to+\infty} \sum_{k=0}^K x_k^\Pi = (I - \Pi^T \cdot P^T)^{-1} \cdot \Pi^T y_0$. ($\lim_{k\to+\infty} (\Pi^T \cdot P^T)^k = 0$ by definition of BT-properness since $\mathbf{1}^T (P^T \cdot \Pi^T)^n \cdot e_i < 1$ for all $i = 1, \ldots, n$). □

Lemma 21. *Let Q be a matrix with $\lim_{k\to+\infty} Q^k = 0$. Then $I - Q$ is invertible, $\Sigma_{k\geq 0} Q^k$ is well defined and $\sum_{k\geq 0} Q^k = (I - Q)^{-1}$.*

A.2. Proof of Proposition 6

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 $\bar{x}(\delta_{G_x}^+(v)) - \bar{x}(\delta_{G_x}^-(v)) = \mathbb{1}_{\{s\}}(v) - \mathbb{1}_{\{0_S\}}(v)$ for all $v \in V(G_x)$ and $\bar{x} \geq 0$. It is thus a unit $(s, 0_S)$-flow in $V(G_x)$. Now let us assume that there exists $s' \in R_{G_x}^+(s)$ with $s' \notin R_{G_x}^-(0)$. Summing up all flow constraints over $v \in R_{G_x}^+(s')$, we get $\bar{x}(\delta_{G_x}^+(R_{G_x}^+(s'))) - \bar{x}(\delta_{G_x}^-(R_{G_x}^+(s'))) = \mathbb{1}_{R_{G_x}^+(i)}(s)$. We have $\bar{x}(\delta_{G_x}^+(R_{G_x}^+(s'))) = 0$ by definition of $R_{G_x}^+(s')$. But then $\bar{x}(\delta_{G_x}^-(R_{G_x}^+(s'))) + \mathbb{1}_{R_{G_x}^+(s')}(s) = 0$. Since $\bar{x}(\delta_{G_x}^-(R_{G_x}^+(s'))) \geq 0$, this implies $s \notin R_{G_x}^+(s')$ and $\bar{x}(\delta_{G_x}^-(R_{G_x}^+(s'))) = 0$. Now because $s \notin R_{G_x}^+(s')$ and $s \in R_{G_x}^-(s')$ (by hypothesis), there is at least one arc of E_x in $\delta_{G_x}^-(R_{G_x}^+(s'))$ but this implies $\bar{x}(\delta_{G_x}^-(R_{G_x}^+(s'))) > 0$ as $\bar{x} > 0$, a contradiction.

References

Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network flows: theory, algorithms, and applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.
Alterovitz, R., Simon, T., & Goldberg, K. (2008). The stochastic motion roadmap: A sampling framework for planning with Markov motion uncertainty. In M. P. W. Burgard, et al. (Eds.), *Proceedings of the robotics: Science and systems III (Proc. RSS 2007)* (pp. 233–241).
Bäuerle, N., & Rieder, U. (2011). *Markov decision processes with applications to finance: Markov decision processes with applications to finance*. Springer Science & Business Media.
Bellman, R. (1957). *Dynamic programming* (1st). Princeton, NJ, USA: Princeton University Press.
Bendali, F., & Quilliot, A. (1990). Réseaux stochastiques. *Revue française d'automatique, d'informatique et de recherche opérationnelle, 24*(2), 167–190.
Bertsekas, D. P. (1991). An auction algorithm for shortest paths. *SIAM Journal on Optimization, 1*(4), 425–447.
Bertsekas, D. P. (2005). Dynamic programming and optimal control. volume I. In *Athena scientific optimization and computation series*. Belmont, MA: Athena Scientific.
Bertsekas, D. P. (2012). Dynamic programming and optimal control. volume II. In *Athena scientific optimization and computation series*. Belmont, MA: Athena Scientific.
Bertsekas, D. P., Pallottino, S., & Scutellá, M. G. (1995). Polynomial auction algorithms for shortest paths. *Computational Optimization and Applications, 4*(2), 99–125.
Bertsekas, D. P., & Tsitsiklis, J. N. (1991). An analysis of stochastic shortest path problems. *Mathematics of Operations Research, 16*(3), 580–595.
Bertsekas, D. P., & Yu, H. (2016). Stochastic shortest path problems under weak conditions. Lab. for Information and Decision Systems Report LIDS-P-2909, MIT.
Denardo, E. V. (1970). On linear programming in a Markov decision problem. *Management Science, 16*(5), 281–288.
d'Epenoux. , F. (1963). A probabilistic production and inventory problem. *Management Science, 10*(1), 98–108.
Eaton, J. H., & Zadeh, L. A. (1962). Optimal pursuit strategies in discrete-state probabilistic systems. *Journal of Basic Engineering, 84*(1), 23–29.
Edmonds, J. (1965). Maximum matching and a polyhedron with (0,1) vertices. *Journal of Research of the National Bureau of Standards, 69*, 125–130.
Fearnley, J. (2010). Exponential lower bounds for policy iteration. . In S. Abramsky, C. Gavoille, C. Kirchner, F. M. a. d. Heide, & P. Spirakis (Eds.), *Automata, languages and programming*. In *Lecture notes in computer science: vol. 6199* (pp. 551–562). Berlin Heidelberg: Springer.
Feinberg, E. A., & Huang, J. (2014). The value iteration algorithm is not strongly polynomial for discounted dynamic programming. *Operations Research Letters, 42*(2), 130–131.
Friedmann, O. (2009). An exponential lower bound for the parity game strategy improvement algorithm as we know it. In *Proceedings of the 24th LICS* (pp. 145–156).
Hansen, T. D. (2012). *Worst-case analysis of strategy iteration and the simplex method*. Aarhus University (Ph.D. thesis).
Hansen, T. D., Miltersen, P. B., & Zwick, U. (2013). Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *Journal of ACM, 60*(1), 1:1–1:16.
Hernández-Lerma, O., & Lasserre, J. B. (2002). The linear programming approach. In E. Feinberg, & A. Shwartz (Eds.), *Handbook of Markov decision processes*. In *International series in operations research & management science: vol. 40* (pp. 377–407). US: Springer.

Please cite this article as: M. Guillot and G. Stauffer, The stochastic shortest path problem: A polyhedral combinatorics perspective, European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2018.10.052

Hordijk, A., & Kallenberg, L. C. M. (1979). Linear programming and Markov decision chains. *Management Science, 25*(4), 352–362.
Howard, R. A. (1960). *Dynamic programming and Markov processes*. New York, London, Cambridge, MA: The MIT press.
Manne, A. S. (1960). Linear programming and sequential decisions. *Management Science, 6*(3), 259–267.
Mausam, & Kolobov, A. (2012). Planning with Markov decision processes: An AI perspective. In *Synthesis lectures on artificial intelligence and machine learning*. Morgan & Claypool Publishers.
Melekopoglou, M., & Condon, A. (1994). On the complexity of the policy improvement algorithm for Markov decision processes. *ORSA Journal on Computing, 6*(2), 188–192.
Merton, R. (1973). An intertemporal capital asset pricing model. *Econometrica, 41*(5), 867–887.
Papadimitriou, C., & Steiglitz, K. (1982). Combinatorial optimization: Algorithms and complexity, Prentice-Hall.
Powell, W. B. (2007). *Approximate dynamic programming: solving the curses of dimensionality (Wiley series in probability and statistics)*. Wiley-Interscience.
Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.
Quilliot, A. (2017). Personal communication.
Shapley, L. S. (1953). Stochastic games. *Proceedings of National Academy of Science, 39*(10), 1095–1100.
Smale, S. (1998). Mathematical problems for the next century. *The Mathematical Intelligencer, 20*(2), 7–15.
Sutton, R. S., & Barto, A. G. (1998). *Introduction to reinforcement learning* (1st). Cambridge, MA, USA: MIT Press.
Teichteil-Königsbuch, F. (2012). Stochastic safest and shortest path problems. In *Proceedings of the twenty-sixth AAAI conference on artificial intelligence, AAAI'12* (pp. 1825–1831). AAAI Press.
White, D. J. (1993). A survey of applications of Markov decision processes. *The Journal of the Operational Research Society, 44*(11), 1073–1096.
Ye, Y. (2005). A new complexity result on solving the Markov decision problem. *Mathematics of Operations Research, 30*(3), 733–749.
Ye, Y. (2011). 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.

Please cite this article as: M. Guillot and G. Stauffer, The stochastic shortest path problem: A polyhedral combinatorics perspective, European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2018.10.052