A Note on Detecting Unbounded Instances of the Online Shortest Path Problem

Stephen D. Boyles*and Tarun Rambha†

December 26, 2015

Abstract

The online shortest path problem is a type of stochastic shortest path problem in which certain arc costs are revealed *en route*, and the path is updated accordingly to minimize expected cost. This note addresses the open problem of determining whether a problem instance admits a finite optimal solution in the presence of negative arc costs. We formulate the problem as a Markov decision process and show ways to detect such instances in the course of solving the problem using standard algorithms such as value and policy iteration.

Keywords: online routing; recourse; stochastic shortest paths; policy iteration; label correcting; absorbing Markov chains; negative arc costs

1 Introduction

The online shortest path (OSP) problem, also known as the shortest path problem with recourse, involves finding a minimum expected cost strategy to reach the destination in a stochastic network in which the arc travel times are revealed *en route* [3, 9, 10]. While positive arc travel times guarantee the existence of an optimal strategy, the presence of negative arc costs[1] can result in strategies with unbounded cost. In such cases, determining if the OSP problem has a finite optimal solution has been an open question since it was first discussed in Provan [9]. In this note, we address this issue by formulating the OSP problem as a

*Assistant Professor, Department of Civil, Architectural, and Environmental Engineering, The University of Texas at Austin
†Ph.D. candidate, Department of Civil, Architectural, and Environmental Engineering, The University of Texas at Austin
[1]For an example of a network with negative arc costs, assume that travelers are offered incentives for using certain arcs.

Markov decision process (MDP) and suggest how unbounded instances can be detected when using standard algorithms such as value and policy iteration.

Consider a probabilistic, directed graph $G = (N, A)$ with a set of nodes N, a set of arcs A, and a destination node v. Assume that a directed path exists from each node to v. Let $FS(i)$ denote the forward star of node i, that is, the set of arcs leaving node i. The cost of arc $(i, j) \in A$ is a discrete random variable $\tilde{c}_{ij}$ taking values from the set $C_{ij} = \{c_{ij}^1, c_{ij}^2, \ldots, c_{ij}^{S_{ij}}\}$. Upon arrival at a node i, the cost of all arcs $(i, j) \in FS(i)$ are revealed, and the traveler chooses the next arc so as to minimize the expected travel cost to the destination. Let t be a dummy node that is connected to the destination, i.e., $(v, t) \in FS(v)$, with an arc cost 0. Also suppose that t has a zero-cost self-loop[2].

Let $\Theta_i = \times_{j:(i,j)\in A} C_{ij}$ represent the set of *node-states* at i, that is, all joint realizations of arc costs in $FS(i)$, and let c_{ij}^{θ} represent the cost of arc (i, j) in node-state $\theta \in \Theta_i$. Finally, let p^{θ} be the probability that node-state θ is observed. The cost of these arcs are determined independently upon each arrival at a node; this implies the "reset" assumption [9], where arcs traversed more than once are not constrained to have the same cost every time.

The OSP problem is to determine the least expected cost from every node i to v, as well as a routing policy $\pi(i, \theta)$ mapping each node $i \in N$ and node-state $\theta \in \Theta_i$ to an adjacent node j. This problem is evidently a total cost MDP, although this connection has not been fully exploited in recent literature on the problem [3, 9, 10]. MDPs provide a framework for sequential decision making in a stochastic environment. Systems that are modeled as MDPs are primarily characterized by a set of states and a set of actions at each state. Upon choosing an action at a particular state, the system transitions to a new state with a certain probability and some cost is incurred. The goal is to find the optimal action to be taken at each state to minimize the total, discounted, or per-stage expected cost. Using this framework, stochastic routing problems can be cast as an MDP by appropriately defining the state space, transition probabilities, and costs associated with choosing an action[3]. For a more comprehensive discussion on MDPs and stochastic shortest paths see Bertsekas and Tsitsiklis [2]. In the remainder of this section, we describe the components of the MDP for the OSP problem and the associated Bellman equations.

For the OSP problem, the states of the MDP are tuples (i, θ) and the action space at state (i, θ) is $\{j \in N : (i, j) \in A\}$, the set of nodes adjacent to i. The cost of choosing $\pi(i, \theta)$ in state (i, θ) is simply $c_{i,\pi(i,\theta)}^{\theta}$. The

[2]The OSP problem with non-negative arc costs may be solved, without defining a dummy node, by initializing the label of the destination to zero. However, when arc costs are negative, a traveler may cycle even after reaching the destination, and hence an absorbing state is necessary.

[3]Examples of other stochastic routing problems include minimizing the expected costs in networks with stochastic, time-varying arc costs [6] and maximizing the probability of arriving on-time [7].

state associated with the dummy node $(t,0)$ forms a cost-free absorbing state as the probability of returning to itself is 1. The value function $\mathcal{L}^{\pi}(i,\theta)$ at state (i,θ) is the expected cost incurred by a traveler at node i and node state θ following policy π. Mathematically, $\mathcal{L}^{\pi}(i,\theta) = \lim_{K\to\infty} \mathbb{E}\left\{\sum_{k=0}^{K} c_{i^k,\pi(i^k,\theta^k)} | i^0 = i, \theta^0 = \theta\right\}$, where (i^k,θ^k) is the state at the k^{th} stage (stages here represent decision points of the traveler). When using policy π, let the expected cost of reaching the destination from node i prior to the realization of node states at i (which we will henceforth refer to as labels) be denoted by L_i^{π} and defined as follows

$$L_i^{\pi} = \sum_{\theta\in\Theta_i} p^{\theta}\mathcal{L}^{\pi}(i,\theta) \tag{1}$$

We refer to L_i^{π} as the cost of the policy π when departing from node i. Denote the set of all policies by Π. The objective is to find $\pi^* \in \Pi$ that minimizes L_i^{π} over the set Π for all $i \in N$. Notice from the definitions of $\mathcal{L}^{\pi}(i,\theta)$ and L_i^{π} that $\mathcal{L}^{\pi}(t,\theta) = L_t^{\pi} = 0$ for $\theta \in \Theta_t$. For all other nodes and $\pi \in \Pi$, the values of $\mathcal{L}^{\pi}(i,\theta)$ satisfy the following equations

$$\mathcal{L}^{\pi}(i,\theta) = c^{\theta}_{i,\pi(i,\theta)} + \sum_{\theta'\in\Theta_{\pi(i,\theta)}} p^{\theta'}\mathcal{L}^{\pi}\left(\pi(i,\theta),\theta'\right) \;\forall\, i \in N\backslash\{t\}, \theta \in \Theta_i \tag{2}$$

Fixing a policy π results in a Markov chain with states (i,θ), where $i \in N, \theta \in \Theta_i$. We will refer to this as the *original Markov chain* for reasons that will become apparent shortly. Let $\mathcal{P}^{\pi}$ represent the matrix of transition probabilities for moving between pairs of states under policy π, that is, for all $i,j \in N\backslash\{t\}, \theta \in \Theta_i, \theta' \in \Theta_j$, $\mathcal{P}^{\pi}\left[(i,\theta),(j,\theta')\right] = p^{\theta'}$ if $\pi(i,\theta) = j$ and is zero otherwise. Using this definition, equation (2) can be compactly written as $\boldsymbol{\mathcal{L}}^{\pi} = \mathbf{c}^{\pi} + \mathcal{P}^{\pi}\boldsymbol{\mathcal{L}}^{\pi}$, where $\boldsymbol{\mathcal{L}}^{\pi}$ and $\mathbf{c}^{\pi}$ are column vectors (of length equal to the total number of states excluding the states at node t) containing the $\mathcal{L}^{\pi}(i,\theta)$ and $c^{\theta}_{i,\pi(i,\theta)}$ values respectively.

Using equations (1) and (2), $\mathcal{L}^{\pi}(i,\theta) = c^{\theta}_{i,\pi(i,\theta)} + L^{\pi}_{\pi(i,\theta)} \;\forall\, i \in N\backslash\{t\}, \theta \in \Theta_i$ and using equation (1) again

$$L_i^{\pi} = \sum_{\theta\in\Theta_i} p^{\theta}\left(c^{\theta}_{i,\pi(i,\theta)} + L^{\pi}_{\pi(i,\theta)}\right) \;\forall\, i \in N\backslash\{t\} \tag{3}$$

The system of equations (3) in matrix form can be written as $\mathbf{L}^{\pi} = \mathbf{b}^{\pi} + P^{\pi}\mathbf{L}^{\pi}$, where $\mathbf{L}^{\pi}$ is a column vector of labels with a component L_i^{π} for each node $i \in N\backslash\{t\}$; $\mathbf{b}^{\pi}$ is a column vector of expected costs with i^{th} element $b^{\pi}[i] = \sum_{\theta\in\Theta_i} p^{\theta} c^{\theta}_{i,\pi(i,\theta)}$; and P^{π} is a transition matrix with elements $P^{\pi}[i,j] = \sum_{\theta\in\Theta_i:\pi(i,\theta)=j} p^{\theta}$ for all $i,j \in N\backslash\{t\}$. The transition matrix corresponds to an *aggregated Markov chain* in which all states that share the same node in the original Markov chain are grouped to form a single state. If a vector of

labels satisfying equation (3) also satisfies

$$\pi(i,\theta) \in \arg\min_{j:(i,j)\in A}\{c_{ij}^{\theta} + L_j^{\pi}\} \qquad \forall i \in N, \theta \in \Theta_i \tag{4}$$

then the labels are said to satisfy the Bellman equations and are optimal to the OSP problem. The reader is encouraged to verify these equations using the example in Appendix A which is based on the network in Figure 1.

When arc costs are allowed to take zero or negative values, the optimal policy may cycle indefinitely without reaching the absorbing state[4]. If the optimal policy in such instances has a bounded cost, then it cycles while incurring zero cost on average. Unbounded problem instances may arise when cycling among a subset of nodes with a negative average cost. Provan [9], using the example in Figure 1, showed that two simple criteria are incapable of precisely distinguishing unbounded instances from those with finite optimal solutions. As shown below, requiring all cycles to have non-negative cost with probability 1 is too strong a condition and excludes well-defined problem instances, while simply forbidding cycles with a negative expected cost is too weak and allows unbounded instances.

First, suppose that arcs $(2,3)$ and $(2,4)$ either have zero cost or a cost $d < 0$ with equal probability. Let node 5 be the destination. If $d = -6$, the cycles $(1,2,3,1)$ and $(1,2,4,1)$ have negative cost with positive probability; yet a finite optimal solution exists: the policy $\pi(2,[0,0,1]) = 5$, $\pi(2,[0,-6,1]) = 4$, $\pi(2,\theta_2) = 3$ for all other θ_2 and labels $\mathbf{L}^{\pi} = \begin{pmatrix} 1 & -2 & 3 & 3 & 0 \end{pmatrix}^T$ satisfy equations (3) and (4).

Next, setting $d = -7$, there are no negative cycles when link costs are replaced by their expected values, yet no solution exists to equations (3) for the policy $\pi(2,[0,-7,1]) = 4$, $\pi(2,\theta_2) = 3$ for all other θ_2. Furthermore, a traveler leaving node 1 will never reach the destination node 5: when following this policy, the travel cost is reduced by an average of $-1/4$ between successive arrivals at node 1, so indefinite cycling can make the travel cost arbitrarily negative even though neither cycle $(1,2,3,1)$ nor cycle $(1,2,4,1)$ has negative expected cost.

This note describes a necessary and sufficient condition separating unbounded instances from instances with a finite optimal solution in the presence of negative costs. We further show that this condition can be naturally checked when solving the OSP problem using standard algorithms for solving MDPs.

[4]While Bertsekas and Tsitsiklis [2] allow non-positive arc costs in their model, they restrict their attention to what they call proper policies which have a positive probability of reaching the destination after at most a certain number of stages.

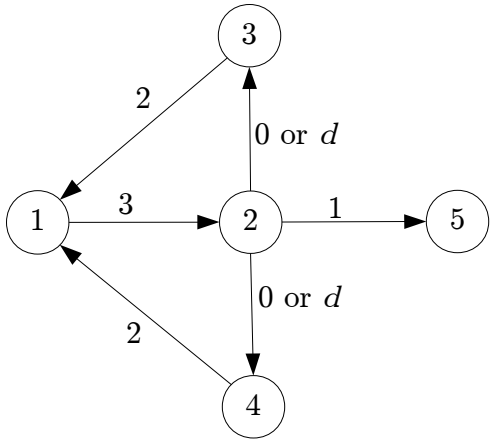

Figure 1: Detecting unbounded problem instances is nontrivial [9].

2 Characterizing policies

We saw earlier that fixing a policy π defines a Markov chain in the original and aggregated state spaces. Notice that these state spaces are finite. In this section, we characterize the states of such Markov chains to understand their long run behavior.

A *closed communicating class* is a subset of states with the property that a Markov chain beginning at any state in the subset reaches every other state in the subset with positive probability, and furthermore remains within the subset with probability one. In general, the state space of a Markov chain can be written as $\mathcal{C}_1 \cup \mathcal{C}_2 \cup \cdots \cup \mathcal{C}_r \cup \mathcal{T}$, where $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r$ are closed communicating classes and $\mathcal{T}$ is the set of transient states. All states in a finite closed communicating class are recurrent and hence we also refer to these classes as recurrent classes.

Clearly the set $\{(t, 0)\}$ is a recurrent class for any policy. Ideally, one would want the optimal policy to yield a Markov chain in which all the remaining states are transient. In such cases, the stochastic process is guaranteed to reach the absorbing state and the expected cost to reach t is finite because the arc costs are bounded and the Markov chain has a finite state space. However, in the presence of non-positive arc costs, other recurrent classes may exist in minimum-expected cost solutions. Thus, our goal is to be able to (i) verify if a given policy contains any recurrent classes and (ii) determine the sign of the cost associated with each recurrent class incurred over an infinite horizon. By a recurrent class we will henceforth refer to a closed communicating class that is not $\{(t, 0)\}$.

Toward the first goal, the following proposition helps us focus our attention on only the aggregated Markov chain and its associated transition matrix P^π. It is advantageous to do so since the size of P^π is

much smaller than that of $\mathcal{P}^\pi$.

Proposition 1. *The original Markov chain has a recurrent class iff the aggregated Markov chain has a recurrent class.*

Proof. ($\Rightarrow$) Let there exist a recurrent class $\mathcal{C}$ containing the state (j, θ_j) in the original Markov chain. To prove that the aggregated Markov chain has a recurrent class, we show that that all states (j, θ'_j), where $\theta'_j \in \Theta_j$, also belong to $\mathcal{C}$. Suppose not, i.e., assume $(j, \theta_j) \in \mathcal{C}$ and $(j, \theta'_j) \notin \mathcal{C}$. Clearly, the recurrent class $\mathcal{C}$ cannot be a singleton containing the state (j, θ_j). Hence, there exists $(i, \theta_i) \in \mathcal{C}$, where $\theta_i \in \Theta_i$, such that $\mathcal{P}^\pi\left[(i, \theta_i), (j, \theta_j)\right] > 0$ and $\pi(i, \theta_i) = j$. Therefore, $\mathcal{P}^\pi\left[(i, \theta_i), (j, \theta'_j)\right] > 0$ for all $\theta'_j \in \Theta_j$, contradicting the fact that (i, θ_i) belongs to $\mathcal{C}$.
($\Leftarrow$) Trivial. ■

Finding recurrent classes in the aggregated Markov chain is easy as one can construct a digraph using the transition matrix and use any reachability algorithm to determine the pairs of states that communicate with each other. The following well-known result gives an alternate way to discover a recurrent class which will be of use later.

Proposition 2. *A policy π contains a recurrent class if and only if the matrix $I - P^\pi$ is singular.*

Proof. If π contains a recurrent class then the system of equations (3) either has no solutions or has infinitely many solutions. In either case $I - P^\pi$ must be singular. To complete the proof, assume that $I - P^\pi$ is singular. Let $\mathcal{C} \subseteq N$ represent the indices of a minimal set of linearly dependent rows of $I - P^\pi$. Then, there exist multipliers μ_i, $i \in \mathcal{C}$, all non-zero, such that $\sum_{i \in \mathcal{C}} \mu_i (I - P^\pi)_i = \mathbf{0}$, where $(I - P^\pi)_i$ is the i^{th} row of $I - P^\pi$. Each multiplier must thus satisfy $\mu_i = \sum_{(h,i): h \in \mathcal{C}} \mu_h P^\pi[h, i]$. Furthermore, one of these multipliers is arbitrary, so fix $\mu_a = 1$, where $a \in \mathcal{C}$. It follows that each multiplier μ_i represents the expected number of times node i is traversed by a random walk starting at a and moving according to policy π, before either returning to a or leaving the set of nodes $\mathcal{C}$. Thus, $\mu_i > 0$ for all $i \in \mathcal{C}$ and every pair of states communicate with each other. Now, consider all columns corresponding to nodes not in $\mathcal{C}$; if $i \in \mathcal{C}$ and k is such a column, $(I - P^\pi)[i, k] = -P^\pi[i, k]$. By linear dependence, we have $\sum_{i \in \mathcal{C}} \mu_i P^\pi[i, k] = 0$. Since $\mu_i > 0$ for all $i \in \mathcal{C}$, $P^\pi[i, k] = 0$ for all i as well. As this holds for all $k \notin \mathcal{C}$, the probability of leaving nodes in $\mathcal{C}$ is zero and hence these nodes form a recurrent class. ■

Let us now address the second goal. If a Markov chain associated with a policy has a recurrent class, the ergodic theorem allows us to use the average cost per stage to distinguish between the bounded and

unbounded instances. Let $P^{\pi}_{\mathcal{C}}$ be the transition matrix of the recurrent class $\mathcal{C}$. In other words, it is a sub-matrix of P^{π} with rows and columns corresponding to the states in $\mathcal{C}$. Then, a non-negative solution to the system of linear equations (5) and (6) gives $\boldsymbol{\lambda}$, the limiting occupancy distribution[5].

$$\lambda_j = \sum_{i \in \mathcal{C}} \lambda_i P^{\pi}_{\mathcal{C}}[i,j] \tag{5}$$

$$\sum_{j \in \mathcal{C}} \lambda_j = 1 \tag{6}$$

Let $\mathbf{b}^{\pi}_{\mathcal{C}}$ be a column vector obtained by selecting the rows of $\mathbf{b}^{\pi}$ that correspond to states in $\mathcal{C}$. Then, the average cost per stage is given by $\boldsymbol{\lambda}^T \mathbf{b}^{\pi}_{\mathcal{C}}$.

For instance, in Figure 1, suppose we follow the policy $\pi(2,[0,-7,1]) = 4$ and $\pi(2,\theta_2) = 3$ for all other θ_2. The limiting occupancy distribution associated with states 1, 2, 3, and 4 is $\lambda = \begin{pmatrix} 1/3 & 1/3 & 1/4 & 1/12 \end{pmatrix}^T$. When $d = -7$, the average cost per stage is $-1/12$. Hence, every transition saves a traveler $-1/12$ units on an average and therefore the problem is unbounded. For $d = -6$, the average cost per stage for the same policy is $1/6$ and is hence not optimal. When $d = -20/3$, the recurrent class has an average cost 0. Existence of a policy containing a recurrent class with negative average cost is necessary and sufficient for OSP to be unbounded below as a traveler starting at any node in such a recurrent class can experience arbitrarily large negative cost by cycling indefinitely.

3 Detecting recurrent classes

Common methods to solve total cost MDPs include policy and value iteration (see Bertsekas [1] for a detailed account of these algorithms and convergence results). In this section, we discuss how to use these methods in networks having non-positive arc costs to address cases in which (i) recurrent classes with zero average cost and (ii) recurrent classes with negative average cost exist. It turns out that the first case can be effortlessly avoided by careful initialization. The second case can also be detected easily using policy iteration, but may be harder to detect when using value iteration.

[5]The limiting occupancy distribution is the fraction of time spent by the system in each of the states of the recurrent class. Note that as the recurrent class is not always aperiodic, $\boldsymbol{\lambda}$ is not necessarily the limiting distribution. For more information on limiting behavior of Markov chains see Kulkarni [5].

3.1 Policy iteration

In policy iteration, we begin with a policy π^0 and use equation (3) to first solve for the labels. Then, using equation (4), a new policy π^1 is derived and these two steps are repeated until the policy remains unchanged between successive iterations. Also, when multiple solutions exist in equation (4), tie-breaking rules are important. For the policy to converge, we set $\pi^{k+1}(i,\theta) = \pi^k(i,\theta)$ if possible (i.e., if $\exists\, j' \neq \pi^k(i,\theta)$ such that $j', \pi^k(i,\theta) \in \arg\min_{j:(i,j)\in A}\{c_{ij}^{\theta} + L_j^{\pi^k}\}$), where k is the iteration number. In order to address case (i) the following proposition is useful.

Proposition 3. *If the policy π^0 used to initialize policy iteration does not contain any recurrent classes with zero or positive average cost per stage, no future iteration will introduce them.*

Proof. To see why, assume that at some iteration k, no recurrent class exists; but at a later iteration $k+1$, a recurrent class exists with node set $\mathcal{C}$. In such cases, we show that the average cost per stage of the recurrent class $\mathcal{C}$ is always negative. Since the policy update is done using $\pi^{k+1}(i,\theta) \in \arg\min_{j:(i,j)\in A}\{c_{ij}^{\theta} + L_j^{\pi^k}\}$, for all nodes in $\mathcal{C}$, we may write

$$c^{\theta}_{i,\pi^{k+1}(i,\theta)} + L^{\pi^k}_{\pi^{k+1}(i,\theta)} \leq c^{\theta}_{i,\pi^{k}(i,\theta)} + L^{\pi^k}_{\pi^{k}(i,\theta)} \;\forall\; i \in \mathcal{C}, \theta \in \Theta_i \tag{7}$$

Recall that if there are ties in the policy update step, we set $\pi^{k+1}(i,\theta) = \pi^k(i,\theta)$ if possible. Further, as $\mathcal{C}$ is a recurrent class in iteration $k+1$ but not in iteration k, the policy must have changed for at least one (i,θ) pair. Therefore, at least one of the above inequalities is strict.

Let the limiting occupancy distribution of recurrent class $\mathcal{C}$ be $\boldsymbol{\lambda}$. For a node $i \in \mathcal{C}$, adding the inequalities (7) for all $\theta \in \Theta_i$, we get

$$\sum_{\theta\in\Theta_i} c^{\theta}_{i,\pi^{k+1}(i,\theta)} \leq \sum_{\theta\in\Theta_i}\left(c^{\theta}_{i,\pi^{k}(i,\theta)} + L^{\pi^k}_{\pi^{k}(i,\theta)} - L^{\pi^k}_{\pi^{k+1}(i,\theta)}\right) \tag{8}$$

$$\sum_{\theta\in\Theta_i} p^{\theta} c^{\theta}_{i,\pi^{k+1}(i,\theta)} \leq \sum_{\theta\in\Theta_i} p^{\theta}\left(c^{\theta}_{i,\pi^{k}(i,\theta)} + L^{\pi^k}_{\pi^{k}(i,\theta)}\right) - \sum_{\theta\in\Theta_i} p^{\theta} L^{\pi^k}_{\pi^{k+1}(i,\theta)} \tag{9}$$

$$b_{\mathcal{C}}^{\pi^{k+1}}[i] \leq L_i^{\pi^k} - \sum_{\theta\in\Theta_i} p^{\theta} L^{\pi^k}_{\pi^{k+1}(i,\theta)} \qquad \text{[using equation (3) and the definition of } \mathbf{b}_{\mathcal{C}}^{\pi}] \tag{10}$$

$$\lambda_i b_{\mathcal{C}}^{\pi^{k+1}}[i] \leq \lambda_i\left(L_i^{\pi^k} - \sum_{\theta\in\Theta_i} p^{\theta} L^{\pi^k}_{\pi^{k+1}(i,\theta)}\right) \qquad \text{[since } \lambda_i \text{ is positive]} \tag{11}$$

Summing the above inequalities for all $i \in \mathcal{C}$ and using the fact that one of the inequalities in (7) is strict,

we have

$$\sum_{i\in\mathcal{C}} \lambda_i b_{\mathcal{C}}^{\pi^{k+1}}[i] < \sum_{i\in\mathcal{C}} \lambda_i \left(L_i^{\pi^k} - \sum_{\theta\in\Theta_i} p^\theta L_{\pi^{k+1}(i,\theta)}^{\pi^k} \right) \tag{12}$$

$$= \sum_{i\in\mathcal{C}} \lambda_i L_i^{\pi^k} - \sum_{i\in\mathcal{C}} \lambda_i \sum_{j\in\mathcal{C}} L_j^{\pi^k} \sum_{\substack{\theta\in\Theta_i: \\ \pi^{k+1}(i,\theta)=j}} p^\theta \tag{13}$$

$$= \sum_{i\in\mathcal{C}} \lambda_i L_i^{\pi^k} - \sum_{j\in\mathcal{C}} L_j^{\pi^k} \sum_{i\in\mathcal{C}} \lambda_i \sum_{\substack{\theta\in\Theta_i: \\ \pi^{k+1}(i,\theta)=j}} p^\theta \tag{14}$$

$$= \sum_{i\in\mathcal{C}} \lambda_i L_i^{\pi^k} - \sum_{j\in\mathcal{C}} L_j^{\pi^k} \lambda_j \qquad \text{[using equation (5)]} \tag{15}$$

$$= 0 \tag{16}$$

Hence, if the policy iteration algorithm discovers a policy with a recurrent class, its average cost per stage can only be negative. ■

Thus, with careful initialization[6], there is no danger of introducing a recurrent class with zero or positive average cost. Therefore, if $I - P^{\pi^k}$ is ever singular, a recurrent class with negative average cost exists and the problem is unbounded.

3.2 Value iteration

Value iteration on the original state space can also be used to solve for the optimal labels. Specifically, in the k^{th} iteration, we estimate $\mathcal{L}^k(i,\theta) = \min_{j:(i,j)\in A} \left\{ c_{ij}^\theta + \sum_{\theta'\in\Theta_j} p^{\theta'} \mathcal{L}^{k-1}(j,\theta') \right\}$ and update the label of node i using $L_i^k = \sum_{\theta\in\Theta_i} p^\theta \mathcal{L}^k(i,\theta)$. If we initialize the values of $\mathcal{L}^0(i,\theta)$ to 0 when $i = t$ and ∞ otherwise, value iteration is known to converge to the optimal labels and the optimal policy can be obtained using equation (4). While this method involves updating all node labels in each iteration, computationally efficient versions (such as the TD-OSP algorithm of Waller and Ziliaskopoulos [10]) which update node labels based on a scan eligible list can also be used. The initialization scheme also ensures that the labels decrease over subsequent iterations and hence value iteration never results in a policy that has a recurrent class with positive average cost. It can however provide an optimal solution with a recurrent class having zero average cost. But in such cases, changing the tie-breaking rules in equation (4) helps discover an optimal policy with no recurrent classes.

[6]For instance, initializing the policy and labels based on a deterministic shortest path problem using expected costs.

Value iteration can converge slowly to the optimal labels which poses a problem when there are recurrent classes with negative average costs. In such instances, as the labels decrease, it is unclear if the labels have not converged or if the problem is unbounded. Lower bounds on the optimal labels can resolve this issue but estimating these bounds appears challenging.

To overcome this issue, a hybrid approach that combines value and policy iteration can be used to solve the OSP problem. Specifically, one could run the TD-OSP algorithm for a fixed number of iterations k_{max}, use the labels to construct a policy, and perform a policy update using equation (3). If $I - P^{\pi}$ is found to be singular during the policy update process, the problem is unbounded. Else, we repeat this process by running the TD-OSP algorithm for another k_{max} iterations followed by a policy update and so on. If, during this process, the scan eligible list in a TD-OSP iteration becomes empty (i.e., an optimal solution is found) or if the policy update step suggests that the problem is unbounded, we terminate.

Alternately, when using the TD-OSP algorithm, one could check if the scan eligible list repeatedly admits a certain sequence of nodes. This would potentially indicate cycling among a subset of nodes. Several methods are available for detecting the reappearance of a set of elements in a sequence. One option, based on Nivasch [8], is to store the scan eligible list from selected iterations in an ordered stack. If a cycle is detected, terminate TD-OSP, construct a policy using the current labels and switch to policy iteration (while making sure that the current policy does not have a recurrent class with zero average cost). Otherwise, TD-OSP will terminate with the optimal solution, and we stop.

To get a sense of the computational effort required for the aforementioned approaches, we solved the OSP problem on the Barcelona network which has 1020 nodes and 2522 links (`http://www.bgu.ac.il/~bargera/tntp/`). These networks are based on standard test instances involving positive costs, which were retained for the experiments in this paper. Since the policy-based algorithms that solve the OSP problem also detect unbounded instances, the computation time for detecting an unbounded instance is expected to be of the same order.

The hybrid approach involved a policy update after every k_{max} TD-OSP iterations. The algorithms were implemented in C and tested on a Linux machine with an 8 core Intel Core i7-870 CPU @ 2.93 GHz. The results are shown in Table 1 and suggest that the run times of all the algorithms are comparable to each other.

Table 1: Average run time (in seconds) for various OSP algorithms.

	Policy Iteration	*Hybrid Approach*					*Value Iteration*
k_{max}	0	$1\|N\|$	$2\|N\|$	$3\|N\|$	$4\|N\|$	$5\|N\|$	–
Time	0.5517	0.5156	0.4835	0.4547	0.4251	0.4472	0.5872

4 Conclusion

This note addressed an open problem concerning the online shortest path problem, identifying a necessary and sufficient condition for the existence of a finite optimal solution in the presence of non-positive arc costs – namely, the nonexistence of a policy with a recurrent class with negative average cost. Viewing the online shortest path problem as a Markov decision process, we discussed methods that are capable of detecting whether the problem admits a finite optimal solution (in which case the algorithm returns the solution), or alternatively discovering a policy with a recurrent class with negative average cost in finite time.

However, there are many open questions surrounding the complexity of the policy iteration method and hence the question of existence of a polynomial-time algorithm for detecting unbounded instances remains to be explored. While it was shown by Ye [11] that policy iteration is strongly polynomial for discounted problems, Fearnley [4] demonstrated, using a carefully constructed example which requires probabilistic actions at some states, that policy iteration for the total and average cost MDP may take an exponential number of steps. Although the OSP problem is a total cost MDP, it is defined on a network, which imposes additional structure that may possibly help in showing that the worst-case instances of policy iteration run in polynomial time.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Nos. 1069141/1157294 and 1254921. Partial support was also provided by the Data-Supported Transportation Planning and Operations University Transportation Center at The University of Texas at Austin. The authors are grateful for this support. The authors would also like to thank two anonymous reviewers for their insightful comments.

References

[1] D.P. Bertsekas, Dynamic programming and optimal control, Vol. II, Athena Scientific, Cambridge, MA, 2007.

[2] D.P. Bertsekas and J.N. Tsitsiklis, An analysis of stochastic shortest path problems, Math Oper Res 16 (1991), 580–595.

[3] R.K. Cheung, Iterative methods for dynamic stochastic shortest path problems, Naval Res Logist 45 (1998), 769–789.

[4] J. Fearnley, "Exponential lower bounds for policy iteration," Automata, languages and programming, S. Abramsky, C. Gavoille, C. Kirchner, F. Meyer auf der Heide, and P. Spirakis (Editors), Springer Berlin Heidelberg, 2010, Vol. 6199 of Lecture Notes in Computer Science, pp. 551–562.

[5] V.G. Kulkarni, Modeling and analysis of stochastic systems, CRC Press, 2009.

[6] E. Miller-Hooks, Adaptive least-expected time paths in stochastic, time-varying transportation and data networks, Networks 37 (2001), 35–52.

[7] Y. Nie and Y. Fan, Arriving-on-time problem: discrete algorithm that ensures convergence, Transportation Res Record: J Transportation Res Board (2006), 193–200.

[8] G. Nivasch, Cycle detection using a stack, Informat Process Lett 90 (2004), 135–140.

[9] J.S. Provan, A polynomial-time algorithm to find shortest paths with recourse, Networks 41 (2003), 115–125.

[10] S.T. Waller and A.K. Ziliaskopoulos, On the online shortest path problem with limited arc cost dependencies, Networks 40 (2002), 216–227.

[11] Y. Ye, The simplex and policy-iteration methods are strongly polynomial for the Markov decision problem with a fixed discount rate, Math Oper Res 36 (2011), 593–603.

Appendix A An example

In this section, we provide an example of the original and aggregated Markov chain defined in Section 1 using the network in Figure 1. Consider the policy $\pi(2,[0,0,1]) = 5$, $\pi(2,[0,d,1]) = 4$, $\pi(2,\theta_2) = 3$ for all other θ_2. The original state space corresponding to policy π is shown in Figure 2. The values on the arcs between states represent the transition probabilities.

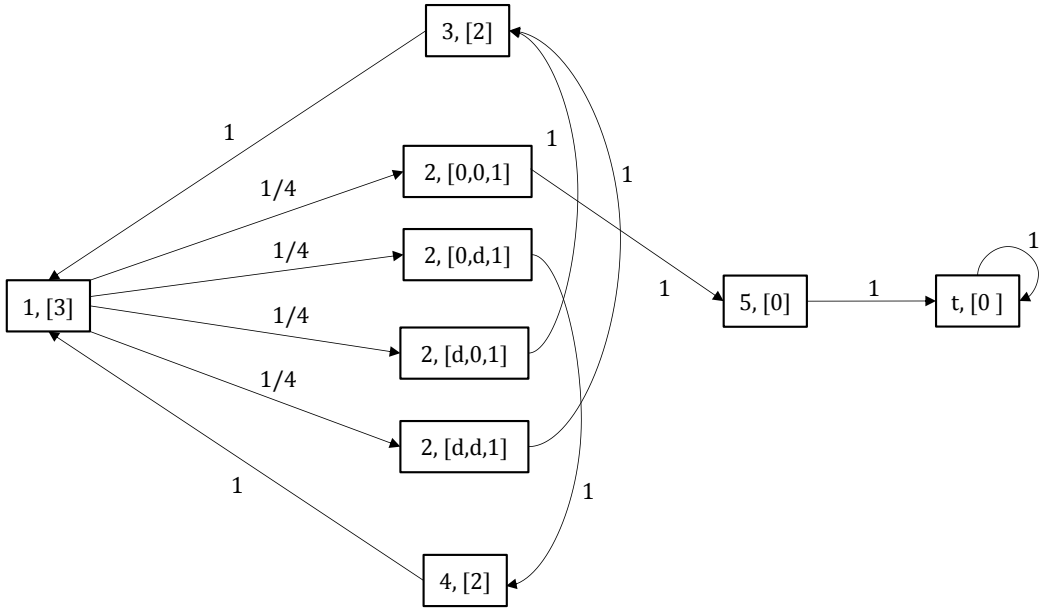

Figure 2: Original state space for policy π.

The transition probability matrix $\mathcal{P}_\pi$ and the cost vector $\mathbf{c}^\pi$ are shown below. The value functions can be obtained by solving $\boldsymbol{\mathcal{L}}^\pi = \mathbf{c}^\pi + \mathcal{P}^\pi \boldsymbol{\mathcal{L}}^\pi$.

$$\mathcal{P}_\pi = \begin{array}{c} \\ 1,[3] \\ 2,[0,0,1] \\ 2,[0,d,1] \\ 2,[d,0,1] \\ 2,[d,d,1] \\ 3,[2] \\ 4,[2] \\ 5,[0] \end{array} \begin{array}{c} \begin{array}{cccccccc} 1,[3] & 2,[0,0,1] & 2,[0,d,1] & 2,[d,0,1] & 2,[d,d,1] & 3,[2] & 4,[2] & 5,[0] \end{array} \\ \begin{pmatrix} 0 & 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \quad \mathbf{c}^\pi = \begin{pmatrix} 3 \\ 1 \\ d \\ d \\ d \\ 2 \\ 2 \\ 0 \end{pmatrix}$$

The aggregated state space groups all states corresponding to node 2 and is shown in Figure 3. The numbers on the arcs represent the transition probabilities defined by P^π.

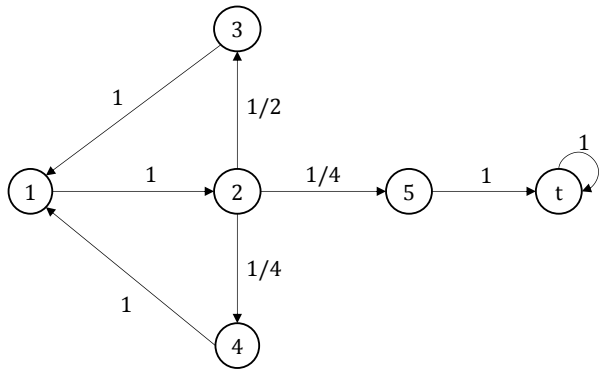

Figure 3: Aggregated state space for policy π.

The labels associated with policy π can be obtained using $\mathbf{L}^\pi = \mathbf{b}^\pi + P^\pi \mathbf{L}^\pi$, where P^π and $\mathbf{b}^\pi$ are shown below. When $d = -6$, solving these equations yields $\mathbf{L}^\pi = \begin{pmatrix} 1 & -2 & 3 & 3 & 0 \end{pmatrix}^T$.

$$P^\pi = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/4 & 1/4 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \qquad \mathbf{b}^\pi = \begin{pmatrix} 3 \\ \frac{d}{2} + \frac{d}{4} + \frac{1}{4} \\ 2 \\ 2 \\ 0 \end{pmatrix}$$