Stochastic Shortest Path with Unlimited Hops ⋆⋆

Ananya Das, Charles Martel
University of California, Davis, CA

Abstract

We present new results for the Stochastic Shortest Path problem when an unlimited number of hops may be used. Nodes and links in the network may be congested or uncongested, and their states change over time. The goal is to adaptively choose the next node to visit such that the expected cost to the destination is minimized. Since the state of a node may change, having the *option* to revisit a node can improve the expected cost. Therefore, the option to use an unbounded number of hops may be required to achieve the minimum expected cost. We show that when revisits are prohibited, the optimal routing problem is NP-hard. We also prove properties about networks for which continual improvement may occur.

We study the related routing problem which asks whether it is possible to determine the optimal next node based on the current node and state, when an unlimited number of hops is allowed. We show that as the number of hops increases, this problem may not converge to a solution.

Key words: online algorithms, graph algorithms, shortest path, stochastic network

1. Introduction

The deterministic Shortest Path Problem is a fundamental problem in computer science. It has been extensively studied and applied in many fields. An interesting version of the problem deals with a stochastic setting in which link costs are *not* known in advance. This problem has recently received much attention and various versions of the problem have been studied [3,4,6–10].

In the stochastic network setting we study, the state of a node is dependent on the states of nearby nodes. The cost of a link depends on the state of its start node. Such a setting was also studied in [4]. The state of a node may change over time and cannot be determined until the node is reached. Therefore, there is uncertainty with both node and link states. To find the cheapest path in such a setting an adaptive routing decision must be made, step-by-step, as the graph is traversed from the source node to the destination node. At each step, the optimal decision for the next step is based on the current position and state, and the knowledge we have about the rest of the graph.

⋆ This work has been supported by NSF Grants No. CNS-0520190 and No. CNS-0832176.
Email addresses: {das}@cs.ucdavis.edu (Ananya Das), {martel}@cs.ucdavis.edu (Charles Martel).

This stochastic shortest path (**SSP**) problem can be applied to many practical settings, such as transportation networks and communication networks. In transportation networks, an accident or natural disaster can cause congestion which will increase travel time. These events often affect a group of nodes and links in a *region* of the network, rather than just one node. Similarly, for many communication networks, hardware failures may occur arbitrarily and lead to congestion in the network. If the service level of a link or node is affected by some random event, it is likely that nearby links or nodes will also be affected.

In this paper, we study the stochastic shortest path problem described above. The authors of [4] studied this problem with the goal of minimizing the expected cost of arriving at the destination from a source. They proposed an iterative solution to solve this problem using a limit on the number of hops. We extend their work in several ways. In [4], it is assumed that the hop limit is $n-1$, where n is the number of nodes in the graph. We consider increasing the hop limit to n or more. We prove that for some networks, the *option* to use n or more hops may improve the expected cost. In these settings, the option to use an infinite number of hops may actually be more beneficial than a finite hop limit. We also prove that the version of the problem that prohibits revisiting a node is NP-hard. We therefore focus on the version where revisiting a node is allowed. Although the optimal solution may then use a large number of hops, it is a practical setting that arises in many communication networks. For example, an adaptive routing approach known as *deflection routing*, is used in some communication networks in which multiple processors send data packets to multiple destinations [1,2,5,11,12]. Contention for communication link access may arise, and is resolved by intentionally deflecting packets. Although this approach may cause some packets to be routed along longer paths, it is useful since little or buffering is required at nodes.

We also study the related routing problem. Given the stochastic network model described above and an unlimited number of hops, this problem asks whether for every step, it is possible to determine which node to visit next to minimize the remaining expected cost. We find that although the optimization problem converges to a solution [4], this decision problem does not.

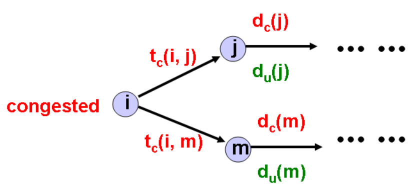
This paper is organized as follows. In Sec. 2 we discuss previously studied versions of **SSP**. In Sec. 3 we discuss our version of the **SSP** problem. In Sec. 4-5 we present our theoretical results. Finally, Sec. 6 provides a summary of our work.

2. Related Work

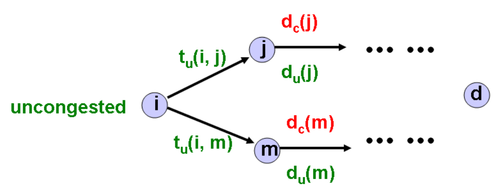
The problem of finding an optimal path in a stochastic network has been studied extensively and various versions of the problem have been considered. In [4], the authors considered a stochastic network in which the cost of links represent travel times. They studied the problem of defining a sequence of nodes to visit such that the minimum expected travel time is achieved. They proposed a dynamic programming solution to find the minimum expected travel time from a source to a destination when the length of the path is limited to a maximum number of hops. In [10], the authors study a network setting in which the state of each link is dependent on the predecessor link, and is independent of the states of nodes. They developed heuristics to define the sequence of links to traverse such that the expected cost is minimized. Minimizing expected cost is also studied in [9]. In this work, the states of links (congested or uncongested) are random, rather than based on a conditional probability. Also, a probability distribution on the duration of congestion is known for every congested link. The authors of [3] considered a different goal. Given a specific deadline, they studied the problem of finding a sequence of choices for the next node that maximizes the probability of arriving on time. In their setting, link travel times are based on probability density functions. They formulate the problem as a system of nonlinear equations, and use approximation schemes to find a good solution. In [8], the authors also consider maximizing the probability of arriving on time. However, in their network model, link travel times are independent random variables drawn from known distributions. In [7], the authors considered a similar setting. The goal of their problem was to achieve the minimum expected travel time and to determine the optimal time to start traveling. The work in [6] considered a setting in which link travel times were dependent on the time of day. They proposed heuristics to estimate the mean and variance of the arrival time for a given destination node.

3. Stochastic Cheapest Cost

In this work, we are interested in a network setting similar to the one studied in [4]. The network consists of nodes and directed links. Each node can be in two possible states: **congested** or **uncongested** (generalizing this assumption to a larger state space is straightforward). The probability that a node is congested is a conditional probability which is based on the states of adjacent nodes. Links in the network are also either congested or uncongested. If we are at a congested node, any outgoing link we traverse will also be in the congested state. Similarly, if we are at an uncongested node, any outgoing link we traverse will be in the uncongested state (see Fig. 1).



(a)Node i is congested.



(b)Node i is uncongested.

Fig. 1. Traversing a graph for the **SSP** problem. (a) Since i is congested, link (i, j) costs $t_c(i, j)$. (b) Since i is uncongested, link (i, j) costs $t_u(i, j)$. In either case, when we arrive at j, if j is congested we visit the node which minimizes $d_c(j)$. If j is uncongested, we visit the node which minimizes $d_u(j)$.

Given this network model, the authors of [4] studied the problem of defining a sequence of nodes to visit that minimizes the expected cost to arrive at the destination node from the source node. We refer to this problem as Stochastic Shortest Path(**SSP**) and now give a formal definition.

3.1. *Problem Statement*

The input is a graph $G = (V, E)$, a source node s, and a destination node d. Every pair of adjacent nodes has associated to it a conditional probability. So, for a pair of adjacent nodes i and j, $P_c(j, i)$ is the probability that node j is congested given that node i is congested. Therefore, if i is congested, then j is uncongested with probability $1 - P_c(j, i)$. Similarly, $P_u(j, i)$ is the probability that node j is uncongested given that node i is uncongested. Every link $(i, j) \in E$ is associated with two costs: $t_c(i, j)$ denotes the congested cost of link (i, j) and $t_u(i, j)$ denotes the uncongested cost of link (i, j). For every link (i, j), $t_c(i, j) \geq t_u(i, j)$. Given such a network, the goal of the **SSP** problem is, starting at s, to adaptively determine the next node to visit that minimizes the remaining expected cost to reach d. Since the states of nodes can change, we cannot determine the optimal path *a priori*. Instead, once we arrive at a node, we must make an adaptive routing decision for the next node to visit.

For example, in Fig. 1(a), since i is congested, traversing link (i, j) will cost $t_c(i, j)$, and in Fig. 1(b), since i is uncongested, traversing (i, j) will cost $t_u(i, j)$. In either case, once we are at node j, if j is congested, then traversing any link from j will cost the congested cost of the link. If j is uncongested, then traversing any link from j will cost the uncongested cost of the link. Once we arrive at j, we find the optimal next node to visit as we did for node i. In [4], the expected costs are expressed as follows:

$$d_c(i) = min\{t_c(i,j) + P_c(j,i) * d_c(j) + (1 - P_c(j,i)) * d_u(j)\} \quad \forall i \ (i \neq j) \tag{1}$$

$$d_u(i) = min\{t_u(i,j) + P_u(j,i) * d_u(j) + (1 - P_u(j,i)) * d_c(j)\} \quad \forall i \ (i \neq j) \tag{2}$$

where $d_c(i)$ is the lowest expected cost from *congested* node i to destination node d, and $d_u(i)$ is the lowest expected cost from *uncongested* node i to destination node d. The authors of [4] proposed a dynamic programming approach to solve equations(1) and(2). They also proved the existence and uniqueness of the solution obtained with their approach. Let $d_c^k(i)$ be the lowest expected cost from congested node i to the destination node d with at most k hops, and let $d_u^k(i)$ be the lowest expected cost from uncongested node i to the destination node d with at most k hops. The following recurrences are presented in [4]:

$$d_c^k(i) = min\{t_c(i,j) + P_c(j,i) * d_c^{k-1}(j) + (1 - P_c(j,i)) * d_u^{k-1}(j)\} \quad \forall i \ (i \neq j) \tag{3}$$

$$d_u^k(i) = min\{t_u(i,j) + P_u(j,i) * d_u^{k-1}(j) + (1 - P_u(j,i)) * d_c^{k-1}(j)\} \quad \forall i \ (i \neq j) \tag{4}$$

Since k is an upper limit on the hop count, the expected cost never increases as k grows larger. The authors of [4] also show that as k grows infinitely large, the $d_c^k(i)$ and $d_u^k(i)$ values converge. This implies that as the hop limit increases, the improvement gained decreases.

Although in the worst case, an infinite number of hops may be required to converge to the true solution, for simplicity, the authors assume that $k \leq n-1$, where n is the number of nodes in G. Since in many realistic network settings, it may be beneficial to allow path lengths longer than $n-1$, we study the effects of increasing k above $n-1$. When $k > n-1$, we know that at least one node has been revisited in the graph. The option to revisit nodes may continuously improve the expected cost. In Sec 4.2 we prove properties about networks for which this continual improvement is possible.

3.2. *Routing Problem*

Using the dynamic programming equations in(3) and(4), for a destination d and some hop limit k, we can determine the optimal next node to visit from each node i, in the congested or uncongested state, using at most k hops. We refer to these values as *next-node* values. For congested node i, we denote the next node value for i using at most k hops as $N_c^k(i)$. For example, from node s in the congested state, the next optimal node to visit using at most 10 hops may be node a. We would denote this as $N_c^{10}(s) = a$. Maintaining these next node values would be a straightforward implementation: as we solve the dynamic programming recurrences in(3) and(4), we would keep track of the node j that minimizes $d_c^k(i)$ and $d_u^k(i)$, respectively.

In some communication networks, contention among data packets for link access is resolved by intentionally deflecting the packets [1,2,5,11,12]. This approach, known as *deflection* routing, is effective when buffers for storing packets are unavailable. To avoid packet loss, the packets are kept moving along the network regardless of the path lengths. For such networks, it may be useful to determine the next node values given an *unlimited* number of hops. We refer to this problem as the *Routing Problem*. Given a destination d, and an unlimited number of hops, the routing decision problem is: given a node i and the state of i, find the optimal next node to visit such that the expected cost from i to d is minimized. In Sec. 3.2 we prove that this problem does not always converge to a solution. In other words, as k grows, the next node values endlessly fluctuate.

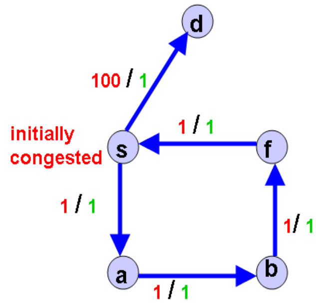


Fig. 2. Example of Continual Improvement. Assume that all conditional probabilities are .5, the congested link from s to d has cost 100, the uncongested link from s to d has cost 1, and all other congested and uncongested links have cost 1. The source is s, which is initially congested, and the destination is d. With $k = 1$, we have $d_c^1(s) = 100$. By traversing the cycle s-a-b-f-s once, we have $d_c^5(s) = 4 + .5{*}1 + .5{*}50 = 54.5$. With the option to visit s again, if it remains congested after 5 hops, we have $d_c^9(s) = 31.75$. The option to continuously revisit s, if s remains congested, will further improve the expected cost.

4. Effects of Revisiting Nodes

Figure 2 shows an example of a network for which the *option* to repeatedly revisit a node may continuously improve the expected cost. Note that, in the figure, if s is ever uncongested, the optimal next step is to visit d. However, while s is congested, the optimal choice is to traverse path s-a-b-f and return to s (with the hope that it is now uncongested). Therefore having the *option* to revisit s will continuously improve the expected cost. We refer to this process of continuously improving the expected cost by revisiting nodes as *continual improvement*.

For networks in which continual improvement is possible, long paths may be required to achieve an expected cost that is close to optimal. We refer to the version of the **SSP** problem for which revisiting a node is prohibited as *Stochastic Shortest Path with No Revisiting*(**SSP-NR**). Unfortunately **SSPNR** is NP-hard (see Sec. 4.1). Therefore we are interested in determining when continual improvement may occur in a network and how much improvement can be gained.

4.1. *Hardness of No Revisits*

Theorem 4.1 *Stochastic Shortest Path with No Revisiting(**SSP-NR**) is* NP*-hard.*
Proof We prove this by reduction from the Hamiltonian Path problem. Let $G = (V, E)$, be an instance of directed Hamiltonian Path, where $n = |V|$. We construct a corresponding instance $G' = (V', E')$ of **SSP-NR** as follows (see Fig. 3):
- $V' = V \cup s \cup d$
- $E' = E \cup (s, i) \cup (i, d)$ for every node $i \in V$
- For every edge $(i, j) \in G$, $t_c(i, j) = 0$ and $t_u(i, j) = 0$ in G'
- For every edge (s, i) in G', $t_c(s, i) = 0$ and $t_u(s, i) = 0$
- For every edge (i, d) in G', $t_c(i, d) = 1$ and $t_u(i, d) = 0$
- For every pair of adjacent nodes i, j in G', $P_c(j, i) = P_u(j, i) = .5$
- Initially, s in G' is congested

The claim is that G has a Hamiltonian path if and only if the expected cost from s to d in G' is $(.5)^n$.

For this input, in any optimal solution for **SSP-NR**, we will start at s and visit non-visited nodes $v_1, v_2, \ldots, v_r$ (where v_r has no arcs to unvisited nodes), such that if v_i is congested, then we will visit v_{i+1}, and if v_i is uncongested, then we will visit d. The expected cost of such a solution will be $.5^r$. If $r = n$, then there must be a Hamiltonian Path in G; otherwise, if $r < n$, then no Hamiltonian

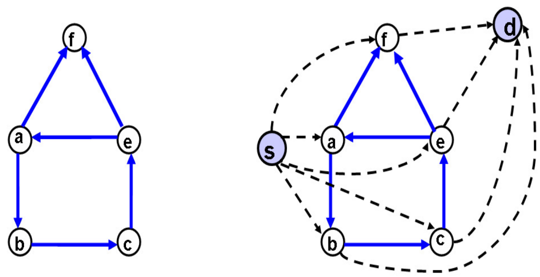


Fig. 3. Reduction from Hamiltonian Path to **SSP-NR**. Graph G (on the left) is an instance of Hamiltonian Path. Graph G' (on the right) is an instance of **SSP-NR**. G' has two additional nodes, s and d. The newly added edges in G' are indicated by dashed lines. Edges which were in G have congested cost=0 and uncongested cost=0 in G'. The newly added edges directed from s have have congested cost=0 and uncongested cost=0. The newly added edges directed to d have congested cost=1 and uncongested cost=0. Since there is a Hamiltonian path in G (a-b-c-e-f), the expected cost from s to d in G' is $.5^5$.

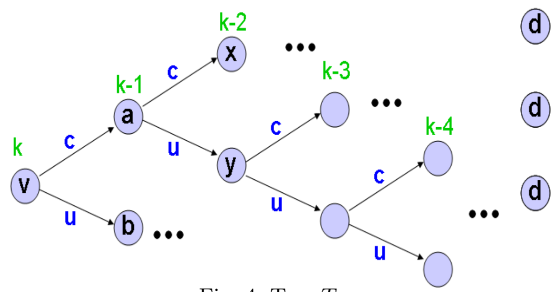


Fig. 4. Tree $T_{v,k}$.

path exists in G.

4.2. *Networks with Continual Improvement*

Since **SSP-NR** is NP-hard, we consider the efficiently solvable version of the problem for which nodes may be revisited. However, as we showed in Fig. 2, with continual improvement an infinite number of potential hops may be required to achieve the optimal expected cost. To solve **SSP** when continual improvement may occur, it may be necessary to limit the number of hops we use. Therefore knowledge about when continual improvement occurs is useful. We now prove that if for some s-d pair, the expected cost achieved using at most $4n$ hops is less than the expected cost using at most $2n$ hops, then continual improvement will occur. For this proof, we use the following notion of a *Tree of Paths*.

Definition *Tree of Paths*: Given a destination d, we can construct a binary tree of paths, $T_{v,k}$, for a node v, and for all positive integers k. $T_{v,k}$ has v as its root and d at all its leaves (see Fig. 4). $T_{v,k}$ is built as follows: We start at the root node v. Let $a = N_c^k(v)$ and $b = N_u^k(v)$, then in $T_{v,k}$, the children of v will be a and b. Similarly, let $x = N_c^{k-1}(a)$ and $y = N_u^{k-1}(a)$. Then in $T_{v,k}$, the children of a will be x and y. Every path in $T_{v,k}$ is the optimal sequence of nodes to visit from v, given the states of the nodes and using at most k hops.

Proposition 4.2 *For $k' > k$, if $d_c^{k'}(v) < d_c^k(v)$, then there is at least one path in $T_{v,k'}$ with length ℓ, such that $k' \geq \ell \geq k + 1$.*

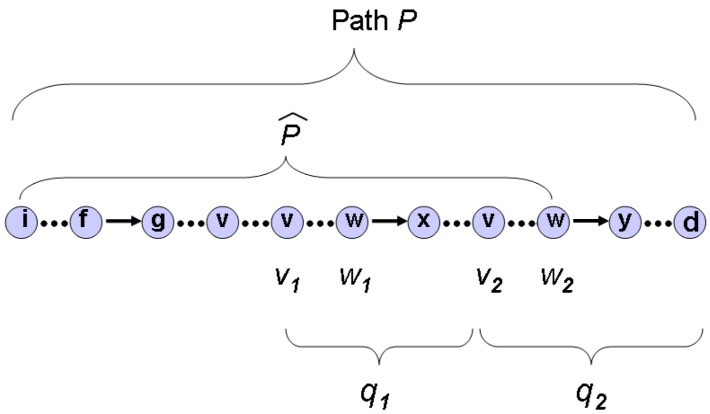


Fig. 5. A path P in $T_{i,2(m+1)n}$ that visits v twice in the same state.

Proof This proof also applies to the case for $d_u^{k'}(v) < d_u^k(v)$. To achieve an expected cost less than $d_c^k(v)$, in $T_{v,k'}$ there must be some path with length more than k hops, otherwise $T_{v,k'}$ would be equivalent to $T_{v,k}$. Therefore, if $d_c^{k'}(v) < d_c^k(v)$, then there must be a path, P, in $T_{v,k'}$ such that $|P| \geq k+1$.

Theorem 4.3 *If for some node i, $d_c^{4n}(i) < d_c^{2n}(i)$, then $\forall\ h \geq 1$,*
$d_c^{2(h+1)n}(i) < d_c^{2hn}(i)$.

Proof This proof also applies to the case for $d_u^{4n}(i) < d_u^{2n}(i)$. The proof is by induction on h.

Base Case: For $h = 1$, we have $d_c^{4n}(i) < d_c^{2n}(i)$, which is true by definition from the proof statement.

Inductive Hypothesis: For some $m \geq 1$, we assume:

$$d_c^{2(m+1)n}(i) < d_c^{2mn}(i)$$

Inductive Step: We will show that $d_c^{2(m+2)n}(i) < d_c^{2(m+1)n}(i)$. From the inductive hypothesis and Prop. 4.2, we know that in $T_{i,2(m+1)n}$, there must be a path P such that $|P| > 2mn \geq 2n+1$. Therefore, on P, there must be some node v that was visited at least 3 times, and for at least 2 of the 3 visits, v must have been in the same state. We denote the last 2 visits made to v in the same state as v_1 and v_2, respectively. Let q_1 be the subpath that starts at v_1 and ends at v_2, and let q_2 be the subpath that starts at v_2 and ends at d (see Fig. 5). Every node along q_1 is associated with a state, so with n nodes, there are at most $2n$ node-state pairs that we can traverse on q_1.

At some node, the sequence of node-state pairs visited along q_1 must differ from the sequence visited along q_2, since $d \notin q_1$. Let x and y denote the first nodes that differ in q_1 and q_2, respectively. Let w denote the node directly before x in q_1, and directly before y in q_2. Without loss of generality, we can assume that w is in the congested state. We denote the visits made to w in q_1 and q_2 as w_1 and w_2, respectively (see Fig. 5). Let $r+\delta$ and r, where $\delta \leq 2n$, denote the number of hops remaining when we reach w_1 and w_2, respectively. We know that $d_c^{r+\delta}(w) < d_c^r(w)$ because if $d_c^{r+\delta}(w) = d_c^r(w)$ then $N_c^{r+\delta}(w) = N_c^r(w)$ [1], and from w_1 we would have gone directly to y instead of x. Therefore the additional δ hops at w_1 allowed us to achieve a lower expected cost.

We will now show that $d_c^{2(m+2)n}(i) < d_c^{2(m+1)n}(i)$. The option of using an additional $2n$ hops will allow a total of at most $2(m+2)n$ hops, so the new tree is $T_{i,2(m+2)n}$. Let $\widehat{P}$ denote the prefix of P in $T_{i,2(m+1)n}$ leading to w_2. Then either: (1) $\widehat{P}$ does not exist in $T_{i,2(m+2)n}$, but instead there is

[1] If two or more choices for the next node yield the minimum expected cost, we break ties by choosing the node that can achieve this minimum cost using the fewest hops.

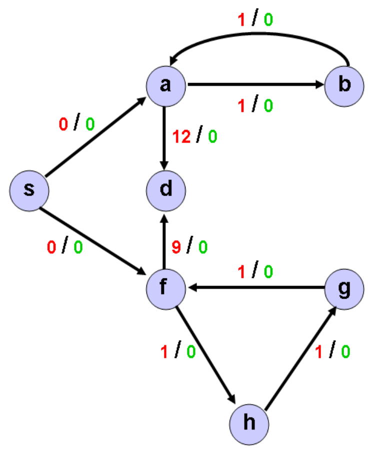


Fig. 6. Example for which the routing decision problem does not stabilize. Link costs (congested followed by uncongested) are shown next to each link.

a different path $\widehat{P}'$ that begins with a prefix of $\widehat{P}$, or (2) $\widehat{P}$ exists in $T_{i,2(m+2)n}$. If (1) occurs, then (see Fig. 5) there must be some nodes f, g, g', such that $\widehat{P}$ and $\widehat{P}'$ are identical up to f, but in $\widehat{P}$ we visit g directly after f, while in $\widehat{P}'$ we visit g' directly after f. Since from f we chose to visit g' instead of g, it must be because this choice improves the expected cost in $\widehat{P}'$ due to now allowing additional hops from f. Therefore, $d_c^{2(m+2)n}(i) < d_c^{2(m+1)n}(i)$. If (2) occurs, then when we arrive at w_2 in $T_{i,2(m+2)n}$ we will have $\delta' \geq r + \delta$ hops remaining. If from w_2 visiting some node z achieves a lower expected cost than visiting x, we will visit z; otherwise, we will visit x. In either case, we will improve the expected cost. Since $d_c^{\delta'}(w) \leq d_c^{r+\delta}(w) < d_c^{r}(w)$, the expected cost for $T_{i,2(m+2)n}$ will be lower than the expected cost for $T_{i,2(m+1)n}$, so $d_c^{2(m+2)n}(i) < d_c^{2(m+1)n}(i)$.

5. Routing Problem Does Not Stabilize

As stated in Sec. 3, the authors of [4] show that for an unlimited number of hops, the **SSP** problem stabilizes. In other words, as k approaches infinity, $d_c^k(i)$ (and $d_u^k(i)$) converges to a limit. In this section, we show that although the **SSP** problem stabilizes, the Routing Problem does not.

Theorem 5.1 *As $k \to \infty$ $N_c^k(i)$ (and $N_u^k(i)$) does not always converge.*

Proof We prove this with an example network shown in Fig. 6. The costs of links (congested followed by uncongested) are shown in the figure. Assume that $P_c(a,s) = P_c(b,a) = P_c(f,s) = P_c(h,f) = P_c(g,h) = 1$, $P_c(a,b) = .5$, and $P_c(f,g) = \frac{1}{3}$. The source s is initially congested, and the destination is d. We now show that for this network, as $k \to \infty$, $N_c^k(s)$ fluctuates between nodes a and f. Since $d_c^7(a) = d_c^7(f) = 5$, if 8 hops are allowed, then visiting a or f from s are equally good choices, so $N_c^8(s) = a$ or $N_c^8(s) = f$. Now, if we increase the number of hops by 2, then traversing the cycle a-b-a will improve the expected cost. If we increase the number of hops by 3, then traversing the cycle f-h-g-f will further improve the expected cost. So with a total of $10, 12, 14, 16, \ldots$ hops, the optimal next node to visit from s is a. With a total of $11, 14, 17, 20, \ldots$ hops, the optimal node to visit from s is f. More specifically, if $k-2$ is divisible by 6, then by using at most k hops, a and f are equally good choices for the next node to visit. Otherwise, when k is even, $N_c^k(s) = a$, and when $k-2$ is divisible by 3, $N_c^k(s) = f$. Therefore, if we continuously increase k by one hop, $N_c^8(s)$ will fluctuate between a and f.

6. Conclusion

We study the problem of minimizing expected cost in a stochastic network. Whereas previous researchers limited the number of hops that can be used [4], we show that by using more hops to revisit nodes, it is possible to continuously improve the expected cost. We also prove that restricting this problem so that nodes may not be revisited is NP-hard. We therefore focus on networks for which continual improvement may occur and prove properties of these networks. Specifically, for an n node graph, we find that if for some source-destination pair, the expected cost achieved using at most $4n$ hops is less than the expected cost using at most $2n$ hops, then continual improvement will occur.

We also study the problem of determining, for every node, the optimal next node to visit to minimize the remaining expected cost. We prove that although the problem of minimizing expected cost stabilizes [4], this routing problem does not.

References

[1] E. Ayanoglu. Signal Flow Graphs for Path Enumeration and Deflection Routing Analysis in Multihop Networks. *IEEE Global Telecommunications Conference*, vol.2, pp. 1022-1029, Nov. 1989.

[2] J. Fàbrega and X. Muñoz. Minimum-Cost QoS Multicast and Unicast Routing in Communication Networks. *Europar 2003 of Lecture Notes in Computer Science*, vol. 2790, pp. 989-994, 2003.

[3] Y. Fan, R. Kalaba, J. Moore. Arriving on Time. *Journal of Optimization Theory and Applications*, vol. 127, no. 3, pp. 497-513, Dec. 2005.

[4] Y. Fan, R. Kalaba, J. Moore. Shortest Paths in Stochastic Networks with Correlated Link Costs. *Computers and Mathematics with Applications*, vol. 49, pp. 1549-1564, 2005.

[5] C. Hsu, T. Liu, N. Huang. Performance Analysis of Deflection Routing in Optical Burst-Switched Networks. *IEEE/ACM Infocom*, vol. 1, pp. 66- 73, 2002.

[6] L. Fu , L. Rillet. Expected Shortest Paths in Dynamic and Stochastic Traffic Networks. *Transportation Research*, vol. 32, no. 7, pp. 499-516, Sep. 1998.

[7] E. Nikolova, M. Brand, and D. Karger. Optimal Route Planning under Uncertainty. *Proceedings of International Conference on Planning and Scheduling*. Lake District, England, June 2006.

[8] E. Nikolova, J. Kelner, M. Brand, M. Mitzenmacher. Stochastic Shortest Paths via Quasi-Convex Maximization. *Proceedings of European Symposium of Algorithms*, Zurich, Switzerland, pp. 552 - 563, 2006.

[9] B. Thomas and C. White. The Dynamic Shortest Path Problem with Anticipation. *European Journal of Operational Research*, vol. 176, no. 2, pp. 836-854, Jan. 2007.

[10] S. Waller, A. Ziliaskopoulos. On the Online Shortest Path Problem with Limited Arc Cost Dependencies. *Networks*, vol, 40, no. 4, pp. 216 - 227, Oct. 2002.

[11] X. Wang, H. Morikawa, and T. Aoyama. Burst Optical Deflection Routing Protocol for Wavelength Routing WDM Networks. *Proceedings of SPIE/IEEE Opticom*, 2000.

[12] A. Zalesky, H. Vu, M. Zukerman, Z. Rosberg, and E. Wong. Evaluation of Limited Wavelength Conversion and Deflection Routing as Methods to Reduce Blocking Probability in Optical Burst Switched Networks. *IEEE IEEE International Conference on Communications*, vol. 3, pp. 1543-1547, June 2004.