Journal of Advances in Mathematics and Computer Science

23(6): 1-10, 2017; Article no.JAMCS.35358

Previously known as British Journal of Mathematics & Computer Science

ISSN: 2231-0851

On Stability of Widest Path in Network Routing

Ahmad Hosseini[1*] and Bita Kabir Baiki[1]

[1]*East Institute of Science and Technology, Tehran, Iran.*

Authors' contributions

This work was carried out in collaboration between both authors. Author AH designed the study, performed the statistical analysis, wrote the protocol and wrote the first draft of the manuscript. Author BKB managed the analysis of the study and the literature search. Both authors read and approved the final manuscript.

Article Information

DOI: 10.9734/JAMCS/2017/35358

Editor(s):

(1) Qiankun Song, Department of Mathematics, Chongqing Jiaotong University, China.

Reviewers:

(1) S. K. Srivatsa, Retired, Anna University, India.

(2) Vaishali D. Khairnar, TEC Under Mumbai University, India.

(3) Zheng Zhao, Zhengzhou Science and Technology Institute, China.

Complete Peer review History: http://www.sciencedomain.org/review-history/20393

Received: 9th July 2017

Accepted: 30th July 2017

Published: 5th August 2017

Original Research Article

Abstract

The problem of *widest path* (WP) is a well-established topic in network routing and digital compositing. This paper contemplates one facet of the robustness of optimal solutions to the widest path; i.e., stability analysis of the WP problem. The study here deals with infimum and supremum perturbations which determine multiplicative changes each individual arc can tolerate conserving the optimality of a given WP. It is additionally illustrated how to determine these marginal values for all arcs, and an algorithm for computing all such values is proposed.

Keywords: Operations research; network routing; path finding; widest path.

2010 Mathematics Subject Classification: 90C90, 68R05, 90B10, 90B06, 90C05.

**Corresponding author: E-mail: ahmad.s.hosseini@gmail.com;*

1 Introduction

Path-finding problems, such as *widest path problem*, is a fundamental component of many important applications in different fields including logistics, networking, shipping, emergency response, project scheduling, cable routing, robotics, message in communication systems, network routing (*digital compositing*, *metabolic pathway analysis*), electrical routing, cycle routing, maximum flow networks, and transportation [1], [2], [3], [4], [5], [6], [7], [8], and [9]. Assuming the relationships between nodes in a network are weighted by a capacity of some sort, the widest path problem involves finding the path between a *source node s* and a *sink node t* that maximizes the minimum capacity in the path [1]. E.g., in a network that models the connections between routers in the Internet, the weight of an arc may represent the *bandwidth* of a connection, and the widest path problem can be employed to find an end-to-end path between two Internet nodes of maximum possible bandwidth. The smallest arc weight on this path is known as the capacity or bandwidth of the path [10].

In many cases the data used are inexact or uncertain. In such cases, stability analysis is necessary to determine the credibility of the optimal solution and conclusions based on that solution. In fact, stability analysis is an important element in making decisions and it investigates the effect of changes on a given optimal solution to a problem. Therefore, such a study can be useful in assessing the 'robustness' of an optimal solution to inaccuracy or variability in the given input data. The basic topic of sensitivity analysis is the special case when the value of a single element is subject to change. The goal of such perturbation is to determine the maximum and minimum additive changes of a given individual weight preserving the optimality of a given optimal solution [11]. Such tolerance calculations have been previously investigated for different problems in different contexts such as transportation, minimum spanning tree, traveling salesman, shortest path, dynamic graph techniques, Vickrey payments, and maximally reliable path [12], [13], [14], and [15].

Related traditional sensitivity analysis problems for *shortest path tree* and MST have been considered initially by [16]. A thirty-year-old result of [17] shows that MST sensitivity analysis can be solved in $O(m\alpha(m,n))$ time, where m is the number of arcs, n the number of nodes, and α the inverse-Ackermann function. Moreover, he gave a linear time reduction from shortest path sensitivity analysis to MST sensitivity analysis. [18] showed that if the arc costs are polynomially bounded in n, then on a unit-cost random access machine the MST verification and sensitivity analysis problems can be solved deterministically in linear time. [14] presented a deterministic algorithm for computing the sensitivity of an MST or shortest path tree in $\mathcal{O}(m \log \alpha(m,n))$ time. This work improves upon the long standing bound of $\mathcal{O}(m\ \alpha(m,n))$. [16] proposed some algorithms for sensitivity analysis of shortest path trees and [19] showed that two of their algorithms can be implemented in $\mathcal{O}(m \log n)$ time and $\mathcal{O}(m)$ space. [20] gave lower bounds on the amount of computation required to update shortest path trees. [21] focused on the algorithmic aspect of computing the Vickrey payments in the context of shortest path routing. They showed that the Vickrey prices (upper tolerances in a shortest path) can be computed in $O(m + n \log n)$ time. For an extensive account on computational issues of tolerances in combinatorial optimization, such as MST problem, minimum Hamiltonian path problem and the traveling salesman problem (TSP), published after 1980, we refer the reader to [11], [22], [23], [24], and [25].

To complement previous works, this paper addresses the issue of stability analysis of WP problem in networks to deal with determining the supremum and infimum multiplicative changes that an arc can tolerate preserving the optimality of a pre-obtained WP. More precisely, we aim to determine how much the arc's capacity can be multiplicatively changed without affecting the WP between two given nodes in the system. We propose an algorithm to do the stability analysis for a given WP restricted to the case when a single arc is changed by a multiplicative factor. The algorithm runs in time $\mathcal{O}(m|P^*|)$ (or even $\mathcal{O}(m)$ if one is only interested in the sup tolerances) instead of $\mathcal{O}(m^2)$ using a naive approach. Examples are also provided to help understand the algorithm and relations.

2 Preliminaries

Let $N = (V, A)$ describe a network, either directed or undirected, with V and $A \subseteq V \times V$ representing the set of *nodes* and *arcs*, respectively. We let $n = |V|$ and $m = |A|$ denote the number of nodes and arcs, respectively. Further, we designate two special nodes, the *source node* $s \in V$ and the *sink node* $t \in V$. Each arc $(i, j) \in A$ has a positive *capacity parameter* $c_{ij} \in (0, +\infty)$ associated with it. The capacity measures the maximum amount of flow that can be transmitted through the arc. A path P from v_1 to v_k is defined by a sequence of nodes $v_1, v_2, \ldots, v_{k-1}, v_k$ with the property that every consecutive pair of nodes v_i and v_{i+1} in the sequence is connected by an arc, more precisely, $P = \bigcup_{i=1}^{k-1}(v_i, v_{i+1})$. The capacity of a (directed) path P is the minimum arc capacity in P. That is, the capacity $C(P)$ of a path P is

$$C(P) = \min_{(i,j)\in P} c_{ij} = \min_{(i,j)\in A} \left\{ c_{ij} x_{ij}^P + M(1 - x_{ij}^P) \right\}, \tag{2.1}$$

where $x_{ij}^P = 1$ if arc (i, j) belongs to path P and zero otherwise, and M is a constant such that $M \geq \max_{(i,j)\in A}\{c_{ij}\}$. The WP problem is dealing with the maximization of function (2.1).

Let $\mathcal{P} = \{P_k\}$ denote the set of all s–t-paths in N, i.e., all paths from source s to sink t. The set $\mathcal{P}$ does not depend on the capacity parameters. We are specifically interested in two subsets of $\mathcal{P}$, namely the sets $\mathcal{P}^+_{(i,j)}$ and $\mathcal{P}^-_{(i,j)}$ that comprise all s–t-paths in N that does and does not include arc (i, j), respectively. We also let $C(\mathcal{P}) := \max_{P\in\mathcal{P}} C(P)$ denote the optimal value for the WP problem, $C\big(\mathcal{P}^+_{(i,j)}\big)$ denote the capacity of a WP in $\mathcal{P}^+_{(i,j)}$, and $C\big(\mathcal{P}^-_{(i,j)}\big)$ denote the capacity of a WP in $\mathcal{P}^-_{(i,j)}$.

In the following sections, we assume that the network is *s*–*t*-*connected* and that a WP is already given/obtained and we are asked to investigate the stability analysis issue with respect to it. It is possible to adapt most shortest path algorithms to compute widest paths, by modifying them to use the bottleneck distance instead of path length. However, in many cases even faster algorithms are possible [1]. To obtain a WP, a *WP algorithm* can be established by modifying, e.g., Dijkstra's algorithm. To do so, we need to initialize the label $d(\cdot)$ of each node to 0 and the source node s to ∞. Further, we update the distance label of a node j if and only if for some node $i \in V$, $(i, j) \in A$ and $d(j) < \min\{d(i), c_{ij}\}$, i.e., we set $d(j) := \max\{d(j), \min\{d(i), c_{ij}\}\}$. The complexity of this algorithm is $\mathcal{O}(m + n \log n)$ for directed networks using a Fibonacci or hollow heap [26] and $\mathcal{O}(m)$ for undirected networks using Thorup's algorithm [27].

3 Widest Path & Stability

This section discusses the multiplicative perturbations for the WP problem on $N = (V, A)$ with a given WP P^*. Here, we establish approaches for computing the inf and sup tolerances of all arcs with respect to P^*. Henceforth, we let $N^{\times\lambda}_{(i,j)}$ denote the network N in which the capacity of arc (i, j) is replaced by λc_{ij} with all other capacity parameters staying unchanged, and $C(N^{\times\lambda}_{(i,j)})$ represents the capacity of the WP in $N^{\times\lambda}_{(i,j)}$. We define the multiplicative inf tolerance $I_{P^*}(i, j)$ and multiplicative sup tolerance $S_{P^*}(i, j)$ of arc (i, j) with respect to P^* as

$$I_{P^*}(i,j) = \inf_{\lambda\in\mathbb{R}} \left\{ \lambda \in (0, 1] \;\middle|\; P^* \text{ is an WP in } N^{\times\lambda}_{(i,j)} \right\}, \tag{3.1}$$

$$S_{P^*}(i,j) = \sup_{\lambda\in\mathbb{R}} \left\{ \lambda \in [1, +\infty) \;\middle|\; P^* \text{ is an WP in } N^{\times\lambda}_{(i,j)} \right\}. \tag{3.2}$$

We remark that the multiplicative inf and sup tolerances *do* depend on a particular WP. For better illustration, let us consider the WP instance presented in Fig. 1. As it is seen, there are three

s–t-paths from $s = 1$ to $t = 6$ among which paths $P^* = P_{1-2-3-6}$ and $P^{**} = P_{1-2-4-6}$ are the network's WPs with capacity 3. It is easy to verify that the following relations hold:

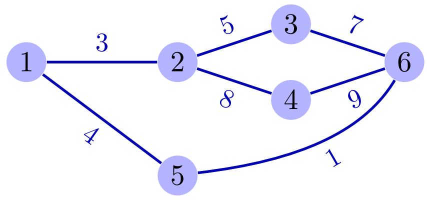

Fig. 1. A WP instance

$$1/3 = I_{P^*}(1,2) = I_{P^{**}}(1,2) = 1/3 \qquad 5/3 = S_{P^*}(1,2) \neq S_{P^{**}}(1,2) = +\infty$$
$$3/5 = I_{P^*}(2,3) \neq I_{P^{**}}(2,3) = 0 \qquad +\infty = S_{P^*}(2,3) = S_{P^{**}}(2,3) = +\infty$$

Property 3.1. ***(a)*** *Let arc $(i,j) \in P^*$, where P^* is an WP in N. Then, $(i,j) \in \bigcap_k P_k \Leftrightarrow I_{P^*}(i,j) = 0$.*
(b) *$(i,j) \notin \bigcup_k P_k \Rightarrow S_{P^*}(i,j) = +\infty$.* □

Note that only the direction "$\Rightarrow$" of Property 3.1(b) holds in general, but not the reverse direction "$\Leftarrow$", which is illustrated in the WP instance presented in Fig. 2. Here, there are three paths from node 1 to node 6 among which the path $P^* = P_{1-2-4-6}$ is the unique WP with capacity of 3. It is easy to see that $S_{P^*}(5,6) = +\infty$, but on the other hand, $(5,6) \in P_{1-5-6}$.

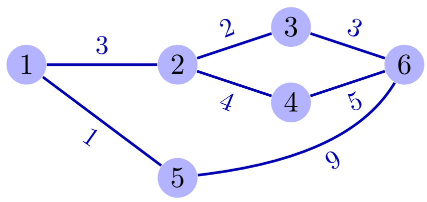

Fig. 2. A WP instance for Property 3.1(b).

Theorem 3.1. *Let $N = (V, A)$ be a WP instance and P^* a WP in N.*

(a) *If $(i,j) \in P^*$, then* $S_{P^*}(i,j) = \begin{cases} +\infty, & \text{if } C(\mathcal{P}) \geq \lim\limits_{\lambda \to +\infty} C\left(N_{(i,j)}^{\times\lambda}\right) \\ \frac{1}{c_{ij}} \min\limits_{(i',j') \in P^* \setminus (i,j)} \{c_{i'j'}\} & \text{if } C(\mathcal{P}) < \lim\limits_{\lambda \to +\infty} C\left(N_{(i,j)}^{\times\lambda}\right) \end{cases}.$

(b) *If $(i,j) \notin P^*$, then* $S_{P^*}(i,j) = \begin{cases} +\infty, & \text{if } \ C(\mathcal{P}) \geq \lim\limits_{\lambda \to +\infty} C\left(N_{(i,j)}^{\times\lambda}\right) \\ \frac{1}{c_{ij}} C(\mathcal{P}) & \text{if } \ C(\mathcal{P}) < \lim\limits_{\lambda \to +\infty} C\left(N_{(i,j)}^{\times\lambda}\right) \end{cases}.$

(c) $C\left(\mathcal{P}_{(i,j)}^{-}\right) = \lim\limits_{\lambda \to 0} C\left(N_{(i,j)}^{\times\lambda}\right)$ *and* $C\left(\mathcal{P}_{(i,j)}^{+}\right) = \max\limits_{P \in \mathcal{P}_{(i,j)}^{+}} \min\limits_{(i',j') \in P} \{c_{i'j'}\}.$

(d) $I_{P^*}(i,j) = \begin{cases} \frac{1}{c_{ij}} C\left(\mathcal{P}_{(i,j)}^{-}\right) & if (i,j) \in P^* \\ 0 & otherwise \end{cases}.$

Proof. For part (a), it should be noted that there can be only two possibilities when dealing with an arc $(i,j) \in P^*$: either P^* will remain a WP for $N^{\times\lambda}_{(i,j)}$ when $\lambda \to +\infty$ or it will not. P^* is a WP for $N^{\times\lambda}_{(i,j)}$ while $\lambda \to +\infty$ if and only if the condition $C(\mathcal{P}) \geq \lim_{\lambda\to+\infty} C\big(N^{\times\lambda}_{(i,j)}\big)$ holds. By definitions of multiplicative sup tolerances, P^* remains a WP under such circumstance, and setting $S_{P^*}(i,j) = +\infty$ is justified. The other case is when P^* is no more a WP in $N^{\times\lambda}_{(i,j)}$ while $\lambda \to +\infty$, and this happens if and only if the condition $C(\mathcal{P}) < \lim_{\lambda\to+\infty} C\big(N^{\times\lambda}_{(i,j)}\big)$ holds. In such situation, in order for P^* to remain a WP for N (justifying the definitions of sup tolerances), any increase more than $\frac{1}{c_{ij}} \min_{(i',j')\in P^*\setminus(i,j)}\{c_{i'j'}\}$ on c_{ij} will create another WP (violating the definitions of sup tolerances). Hence, our settings are validated. For illustrative proof, we refer to Figs. 1-5.

The proof for Part (b) is established along the same lines as in the proof for Part (a). Part (c) is established by considering the fact that the WP instance $N^{\times\lambda}_{(i,j)}$ actually represents $N\,(V, A\setminus(i,j))$ when λ approaches zero. For Part (d), let $(i,j) \in P^*$. By the definition of inf tolerance (3.1), we have

$$I_{P^*}(i,j) = \inf_{\lambda\in\mathbb{R}} \big\{\lambda \in (0,1] \mid \lambda c_{ij} \geq C\big(\mathcal{P}^-_{(i,j)}\big)\big\} = \frac{1}{c_{ij}} C\big(\mathcal{P}^-_{(i,j)}\big). \tag{3.3}$$

□

By employing Theorem 3.1, we can calculate the exact values of all multiplicative inf and sup tolerances of an arc (i,j) with respect to a given WP in the same asymptotic time complexity as at most two WP algorithms. Therefore, the total computational effort will be $\mathcal{O}(m^2 + mn\log n)$ for directed networks and $\mathcal{O}(m^2)$ for undirected networks. However, we now use the previous results to develop the following WP-IST algorithm for computing the tolerances of all arcs in a reduced computational time. To proceed, we need to define an auxiliary network, called *residual network*.

Let $N = (V,A)$ be an WP instance and let P be an arbitrary s–t-path in N with capacity $C(P)$. The residual network with respect to path P is $N^r_P = (V, A^r_P)$, where $A^r_P = \{(i,j) \in A \mid c_{ij} > C(P)\}$. Consequently, $N^r_{P\setminus(k,l)} = \big(V, A^r_{P\setminus(k,l)}\big)$, where $A^r_{P\setminus(k,l)} = \{(i,j) \in A \mid c_{ij} > C(P\setminus(k,l))\}$. It is easy to see that with respect to some WP P^* that the residual network $N^r_{P^*} = (V, A^r_{P^*})$ is an *s–t-disconnected network*. More precisely, the node set V can be partitioned into at least two disconnected components. Therefore, we define a possible cut. Let V_s denote the set of nodes reachable from s in $N^r_{P^*}$ and V_t be the set of nodes that are reachable from t in $N^r_{P^*}$, and let V_s and V_t be non-empty. Then in an undirected network, define $\mathrm{Cut}\,(N^r_{P^*}, V_s, V_t)$ as the set of pairs (i,j) satisfying either that $i \in V_s$ and $j \in V_t$ or $i \in V_t$ and $j \in V_s$. Similarly, in a directed network define $\mathrm{Cut}\,(N^r_{P^*}, V_s, V_t)$ as the set of pairs (i,j) satisfying that $i \in V_s$ and $j \in V_t$. $\mathrm{Cut}\,(N^r_{P^*}, V_s, V_t)$ can also be translated in the original network N. Namely, V_s consists of all nodes i for which there is a path $P_{s\to i}$ whose capacity is strictly larger than $C(\mathcal{P})$ in N, and V_t contains all nodes j from which there is a path $P_{j\to t}$ whose capacity is strictly larger than $C(\mathcal{P})$ in N.

Assuming that a WP P^* is already obtained, we perform the *WP-IST Algorithm* to efficiently calculate all arcs' tolerances for an WP instance $N = (V,A)$. We give the algorithm in a pseudo code (WP-IST Alg.) which runs in $\mathcal{O}(m)$ (if only the sup tolerances are concerned) or $\mathcal{O}(|P^*|m)$ time (if both the sup and inf tolerances are concerned). The WP-IST algorithm originally was developed to calculate the sup tolerances, however, it is capable to calculate the inf tolerances also.

WP-IST Algorithm

Step 0: Preparation

The WP instance $N = (V, A)$ is at hand. So is a WP P^*

Step 1: Construction

1.1. Set $(k, l) = \arg\min_{(i,j)\in P^*}\{c_{ij}\}$ (note: (k, l) may not be unique)
1.2. Set $C(\mathcal{P}) = C(P^*) = \min_{(i,j)\in P^*}\{c_{ij}\}$
1.3. Construct $N^r_{P^*} = (V, A^r_{P^*})$ and $\mathrm{Cut}\,(N^r_{P^*}, V_s, V_t)$ on $N^r_{P^*}$
1.4. Set $C\,(P^* \setminus (k, l)) = \min_{(i,j)\in P^*\setminus(k,l)}\{c_{ij}\}$
1.5. Construct $N^r_{P^*\setminus(k,l)} = \left(V, A^r_{P^*\setminus(k,l)}\right)$ and $\mathrm{Cut}\left(N^r_{P^*\setminus(k,l)}, V_s, V_t\right)$ on $N^r_{P^*\setminus(k,l)}$

Step 2: Search over arc set $A \setminus P^*$

2.1. For $(i, j) \in A$ and $(i, j) \notin P^*$ set $I_{P^*}(i, j) = 0$
2.2. For $(i, j) \in A$, $(i, j) \notin P^*$, and $(i, j) \in \mathrm{Cut}(N^r_{P^*}, V_s, V_t)$ set $S_{P^*}(i, j) = \frac{1}{c_{ij}} C(\mathcal{P})$
2.3. For $(i, j) \in A$, $(i, j) \notin P^*$, and $(i, j) \notin \mathrm{Cut}\,(N^r_{P^*}, V_s, V_t)$ set $S_{P^*}(i, j) = +\infty$

Step 3: Search over arc set $A \cap P^*$

3.1. For $(i, j) \in A$ and $(i, j) \in P^*$ set $I_{P^*}(i, j) = \frac{1}{c_{ij}} C\big(\mathcal{P}^-_{(i,j)}\big)$
3.2. For $(i, j) \in A$, $(i, j) \in P^*$, and $c_{ij} > C(\mathcal{P})$ set $S_{P^*}(i, j) = +\infty$
3.3. For $(i, j) \in A$, $(i, j) \in P^*$, $c_{ij} = C(\mathcal{P})$ and $(i, j) \in \mathrm{Cut}\left(N^r_{P^*\setminus(k,l)}, V_s, V_t\right))$ set $S_{P^*}(i, j) = \frac{1}{c_{ij}} \min_{(i',j')\in P^*\setminus(k,l)}\{c_{i'j'}\}$
3.4. For $(i, j) \in A$, $(i, j) \in P^*$, $c_{ij} = C(\mathcal{P})$, and $(i, j) \notin \mathrm{Cut}\left(N^r_{P^*\setminus(k,l)}, V_s, V_t\right)$ set $S_{P^*}(i, j) = +\infty$

At Step 2, the algorithm makes use of $\mathrm{Cut}\,(N^r_{P^*}, V_s, V_t)$ for any arc $(i, j) \in A \setminus P^*$ to determine the arcs whose capacities' changes impact the optimality of P^*. Those arcs are exactly the ones that belong to $\mathrm{Cut}\,(N^r_{P^*}, V_s, V_t) \setminus P^*$ and were discussed in Theorem 3.1. Note that any arc $(i, j) \in \mathrm{Cut}\,(N^r_{P^*}, V_s, V_t)$ can be a *bottleneck arc* whose capacity's increase may affect the optimality of the already obtained WP, because it may create a path of capacity strictly larger than $C(P^*) = C(\mathcal{P})$. Having detected those bottleneck arcs, we correctly set the tolerances' values for all arcs $(i, j) \in A \setminus P^*$ using the results of corresponding properties and theorems established previously.

Analogously, at Step 3, the algorithm sets the inf tolerances for all arcs $(i, j) \in A \cap P^*$ again using the results of Theorem 3.1. Then it sets the sup tolerances for arcs $(i, j) \in A \cap P^*$ whose capacities are larger than $C(N)$. A closer scrutiny and another use of Theorem 3.1 reveal that the arcs in $A \cap P^*$ that may affect the optimality of P^* are exactly those belonging to $\mathrm{Cut}(N^r_{P^*\setminus(k,l)}, V_s, V_t)$ with capacity $C(\mathcal{P})$. In other words, any arc $(i, j) \in A \cap P^*$ with $c_{ij} = C(\mathcal{P})$ can be a bottleneck arc. Indeed, arc $(i, j) \in \mathrm{Cut}(N^r_{P^*\setminus(k,l)}, V_s, V_t) \cap P^*$ whose capacity is $C(\mathcal{P})$ can create a better WP, so we should limit it (by sup tolerance) using Theorem 3.1.

Taking the fact $(A \backslash P^*) \bigcup (A \cap P^*) = A$ into account, the algorithm determines the multiplicative sup tolerances of all arcs in $\mathcal{O}(m)$ time. Therefore, if only the sup tolerances are required, the running time of the WP-IST algorithm is $\mathcal{O}(m)$, which outperforms the naive $\mathcal{O}(m^2)$-implementation discussed earlier. If both inf and sup tolerances are concerned, the complexity is $\mathcal{O}(m) + \mathcal{O}(|P^*|m) = \mathcal{O}(|P^*|m)$. The *bottleneck operation* of the algorithm is the scanning of arcs $(i, j) \in A \cap P^*$ at Step 3.1, which takes $\mathcal{O}(|P^*|m)$ time. The algorithm performs the construction step in $\mathcal{O}(m)$ time.

Example 3.1. Let us consider the WP instance presented in Fig. 3. There exist several s–t-paths from $s = 1$ to $t = 9$ among which the path $P^* = P_{1-5-8-9}$ is a WP of capacity 5. It can be seen that

$$(k, l) = (8, 9) = \arg \min_{(i',j') \in P^*} \{c_{i'j'}\} \qquad C(\mathcal{P}) = C(P^*) = 5, \tag{3.4}$$

$$\arg \min_{(i',j') \in P^* \setminus (k,l)} \{c_{i'j'}\} = (1, 5) \qquad C\big(P^* \setminus (k, l)\big) = 6. \tag{3.5}$$

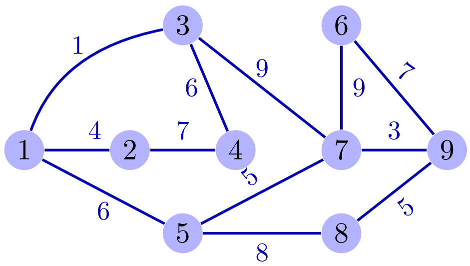

Fig. 3. A WP instance $N = (V, A)$ for the WP-IST algorithm.

Having this information at hand, we can now construct the residual networks as follows:

$$N^r_{P^*} = (V, A^r_{P^*}), \qquad \text{where} \quad A^r_{P^*} = \{(i, j) \in A \mid c_{ij} > 5\}\,, \tag{3.6}$$

$$N^r_{P^* \setminus (k,l)} = \big(V, A^r_{P^* \setminus (k,l)}\big)\,, \qquad \text{where} \quad A^r_{P^* \setminus (k,l)} = \{(i, j) \in A \mid c_{ij} > 6\}\,. \tag{3.7}$$

Finally, employing the algorithm over sets $(A \backslash P^*)$ and $(A \cap P^*)$, we can calculate all arcs' tolerances. To this end, we need the following quantities:

$$C\big(\mathcal{P}^-_{(1,5)}\big) = 4, \qquad C\big(\mathcal{P}^-_{(5,8)}\big) = 5, \qquad C\big(\mathcal{P}^-_{(8,9)}\big) = 5.$$

We denote the multiplicative inf and sup tolerances of each arc (i, j) by *tolerance interval* $[I_{P^*}(i, j), S_{P^*}(i, j)]$ in the network (see Fig. 5).

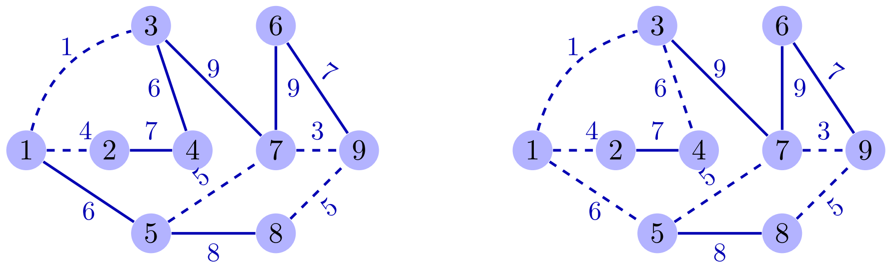

Fig. 4. The residual network $N^r_{P*} = (V, A^r_{P*})$ (left side) and the residual network $N^r_{P*\setminus(k,l)} = (V, A^r_{P*\setminus(k,l)})$ (right side).

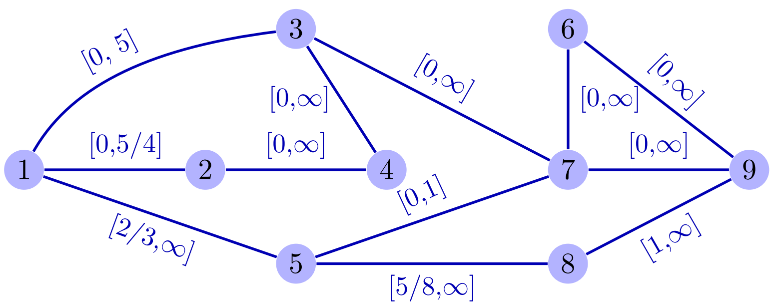

Fig. 5. The WP instance of Fig. 3. with all arcs' multiplicative inf and sup tolerances.

As examples, we show the calculations and considerations taken by the algorithm. The validity of the obtained values can be checked by Theorem 3.1.

$$
\begin{aligned}
& V_s = \{1\} \text{ and } V_t = \{3,6,7,9\} \\
& \text{Cut}\left(N^r_{P*\setminus(k,l)}, V_s, V_t\right) = \{(1,3)\}: \\
& (1,5) \in P^* \Rightarrow I_{P^*}(1,5) = 2/3, \qquad c_{15} > C(\mathcal{P}) = 5 \Rightarrow S_{P^*}(1,5) = \infty, \\
& (5,8) \in P^* \Rightarrow I_{P^*}(5,8) = 5/8, \qquad c_{58} > C(\mathcal{P}) = 5 \Rightarrow S_{P^*}(5,8) = \infty, \\
& (8,9) \in P^* \Rightarrow I_{P^*}(8,9) = 5/5, \qquad c_{89} = C(\mathcal{P}) = 5 \;\&\; (8,9) \notin \text{Cut}\left(N^r_{P*\setminus(k,l)}, V_s, V_t\right) \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \Rightarrow S_{P^*}(8,9) = \infty, \\
& V_s = \{1,5,8\} \text{ and } V_t = \{2,3,4,6,7,9\} \\
& \text{Cut}\left(N^r_{P*}, V_s, V_t\right) = \{(1,2),(1,3),(5,7),(8,9)\}: \\
& (1,2) \notin P^* \Rightarrow I_{P^*}(1,2) = 0, \qquad (1,2) \in \text{Cut}\left(N^r_{P*}, V_s, V_t\right) \Rightarrow S_{P^*}(1,2) = 5/4, \\
& (1,3) \notin P^* \Rightarrow I_{P^*}(1,3) = 0, \qquad (1,3) \in \text{Cut}\left(N^r_{P*}, V_s, V_t\right) \Rightarrow S_{P^*}(1,3) = 5, \\
& (5,7) \notin P^* \Rightarrow I_{P^*}(5,7) = 0, \qquad (5,7) \in \text{Cut}\left(N^r_{P*}, V_s, V_t\right) \Rightarrow S_{P^*}(5,7) = 1.
\end{aligned}
$$

For arcs $\{(2,4),(3,4),(3,7),(6,7),(6,9),(7,9)\}$, it holds that they are not in P^* and thus their inf tolerance is set to zero. Moreover, they are not in $\text{Cut}(N^r_{P*}, V_s, V_t)$, and thus their sup tolerance is set to ∞.

4 Summary & Discussion

We addressed the issue of stability analysis to deal with infimum and supremum multiplicative perturbations in widest path (WP) problem in network routing . We proposed an algorithm to do the WP stability analysis when an arc is changed by a multiplicative factor. We however believe that there is room for further improvement in our algorithm, in particular, Step 3. Moreover, one can also take advantage of the algorithm's inherent parallelism at Step 3.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Ahuja RK, Magnanti TL, Orlin JB. Network Flows: Theory, algorithms, and applications. Prentice Hall, Upper Saddle River (N.J.); 1993.

[2] Ullah E, Lee K, Hassoun S. An algorithm for identifying dominant-edge metabolic pathways. In: Computer-Aided Design-Digest of Technical Papers, 2009. ICCAD 2009. IEEE/ACM International Conference on. IEEE. 2009;144-150.

[3] Hosseini A. An introduction to dynamic generative networks: Minimum cost flow. Applied Mathematical Modelling. 2011;35(10):5017-5025.

[4] Schulze M. A new monotonic, clone-independent, reversal symmetric, and condorcet-consistent single-winner election method. Social Choice and Welfare. 2011;36(2):267-303.

[5] Berman O, Handler GY. Optimal minimax path of a single service unit on a network to nonservice destinations. Transportation Science. 1987;21(2):115-122.

[6] Dietrich B, Vohra R. Mathematics of the Internet. Prentice Hall, Upper Saddle River; 1993.

[7] Hosseini A. Time-dependent optimization of a multi-item uncertain supply chain network: A hybrid approximation algorithm. Discrete Optimization. 2015;18:150-167.

[8] Fernandez E, Garfinkel R, Arbiol R. Mosaicking of aerial photographic maps via seams defined by bottleneck shortest paths. Operations Research. 1998;46(3):293-304.

[9] Hosseini A, Sahin G, Unluyurt T. A penalty-based scaling algorithm for the multi-period multi-product distribution planning problem. Engineering Optimization. 2017;49(4):583-596.

[10] Shacham N. Multicast routing of hierarchical data. In: Communications, 1992. ICC'92, Conference record, SUPERCOMM/ICC'92, Discovering a New World of Communications., IEEE International Conference on. IEEE. 1992;1217-1221.

[11] Gal T. Sensitivity analysis, parametric programming, and related topics: Degeneracy, multicriteria decision making redundancy. Walter de Gruyter, Berlin, Germany; 1995.

[12] Muller-Merbach H. Sensibilitatsanalyse von Transportproblemen der linearen Planungsrechnung (mit ALGOLProgramm); 1968.

[13] Elmaghraby SE. Sensitivity analysis of multiterminal flow networks. Operations Research. 1964;12(5):680-688.

[14] Pettie S. Sensitivity analysis of minimum spanning trees in sub-inverse-ackermann time. In: International Symposium on Algorithms and Computation. Springer. 2005;964-973.

[15] Hosseini A, Kabir Baiki B. An addendum on postoptimality of maximally reliable path. Journal of Mathematics Research. 2017;9(3):23-29.

[16] Shier D, Witzgall C. Arc tolerances in shortest path and network ow problems. Networks. 1980;10(4):277-291.

[17] Tarjan RE. Sensitivity analysis of minimum spanning trees and shortest path trees. Information Processing Letters. 1982;14(1):30-33.

[18] Harel D. A linear algorithm for finding dominators in flow graphs and related problems. In: Proceedings of the seventeenth annual ACM symposium on Theory of computing. ACM. 1985; 185-194.

[19] Gusfield D. A note on arc tolerances in sparse shortest-path and network flow problems. Networks. 1983;13(2):191-196.

[20] Spira PM, Pan A. On finding and updating spanning trees and shortest paths. SIAM Journal on Computing. 1975;4(3):375-380.

[21] Hershberger J, Suri S. Vickrey prices and shortest paths: What is an edge worth? In: Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on. IEEE. 2001;252-259.

[22] Hsu L-H, Jan R-H, Lee Y-C, Hung C-N, Chern M-S. Finding the most vital edge with respect to minimum spanning tree in weighted graphs. Information Processing Letters. 1991;39(5):277-281.

[23] Libura M. Sensitivity analysis for minimum hamiltonian path and traveling salesman problems. Discrete Applied Mathematics. 1991;30(2-3):197-211.

[24] Venema S, Shen H, Suraweera F. NC algorithms for the single most vital edge problem with respect to all pairs shortest paths. Information Processing Letters. 2000;10(1):51-58.

[25] Kurzhanski A, Varaiyab P. On reachability under uncertainty. SIAM Journal on Control and Optimization. 200641(1):181-216.

[26] Hansen T, Kaplan H, Tarjan R, Zwick U. Hollow heaps. In: Proceedings of ICALP 2015. Vol. 9134 of Lecture Notes In Computer Science. Springer. 2015;689-700.

[27] Thorup M. Undirected single source shortest paths in linear time. In: 38th Annual Symposium on the Foundations of Computer Science. Miami Beach. 1997;12-21.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://sciencedomain.org/review-history/20393