J Comb Optim
DOI 10.1007/s10878-007-9120-2

Finding the anti-block vital edge of a shortest path between two nodes

Bing Su · Qingchuan Xu · Peng Xiao

Abstract Let $P_G(s,t)$ denote a shortest path between two nodes s and t in an undirected graph G with nonnegative edge weights. A detour at a node $u \in P_G(s,t) = (s,\dots,u,v,\dots,t)$ is defined as a shortest path $P_{G-e}(u,t)$ from u to t which does not make use of (u,v). In this paper, we focus on the problem of finding an edge $e=(u,v) \in P_G(s,t)$ whose removal produces a detour at node u such that the ratio of the length of $P_{G-e}(u,t)$ to the length of $P_G(u,t)$ is maximum. We define such an edge as an anti-block vital edge (AVE for short), and show that this problem can be solved in $O(mn)$ time, where n and m denote the number of nodes and edges in the graph, respectively. Some applications of the AVE for two special traffic networks are shown.

Keywords Edge failures · Shortest path · Anti-block vital edge

1 Introduction

Suppose that in a given transportation network, each edge is associated with a number specifying time needed in a traverse of the edge. In practice, the network is unreliable,

This research is supported by NSF of China under Grants 70471035, 70525004, 701210001 and 60736027, and PSF of China under Grant 20060401003.

B. Su (✉) · Q. Xu · P. Xiao
School of Management, Xi'an JiaoTong University, Xi'an, Shaanxi 710049, China
e-mail: subing@mail.xjtu.edu.cn

Q. Xu
e-mail: xuqch@mail.xjtu.edu.cn

P. Xiao
e-mail: xiaopeng@mail.xjtu.edu.cn

B. Su · Q. Xu · P. Xiao
The State Key Lab for Manufacturing Systems Engineering, Xi'an, Shaanxi 710049, China

some roads may be unavailable at certain times (e.g. blocked by unexpected events such as traffic accidents or snowfall). From the transportation network management point of view, it is valuable to evaluate the inefficiency caused by the failure of a component.

In the past, the problem of an edge removal results in the increase of the length of the shortest path between two nodes in a graph has been studied. Corley and Sha (1982) studied the MVE problem of finding an edge whose removal resulting in the largest increase in the length of the shortest path between two given nodes. This edge is generally referred as the most vital edge with respect to the shortest path. This problem has been solved efficiently by Malik et al. (1989), who gave an $O(m + n \log n)$ time algorithm by using Fibonacci heap. Nardelli et al. (2001) improved the previous time bound to $O(m \cdot \alpha(m, n))$, where α is the functional inverse of the Ackermann function, and n and m denote the number of nodes and edges in the graph, respectively.

Nardelli et al. (1998) focused on the LD (Longest Detour) problem of finding an edge $e^* = (u^*, v^*)$ in the shortest path between two given nodes such that when this edge is removed, the length of the detour satisfies $d_{G-e^*}(u^*, t) - d_G(u^*, t) \geq d_{G-e}(u, t) - d_G(u, t)$, where $G - e = (V, E - e)$. The edge whose removal will result in the longest detour is named the detour-critical edge. They showed that this problem can be solved in $O(m + n \log n)$ time in undirected graphs. The same bound for undirected graphs was also achieved by Hershberger and Suri (2001), who solved the Vickrey payment problem with an algorithm that also solved the detour problem. Nardelli et al. (2001) improved the result to $O(m \cdot \alpha(m, n))$ time bound. Hershberger et al. (2003) showed that the detour problem required $\Omega(m\sqrt{n})$ time in the worst case whenever $m = n\sqrt{n}$ in directed graphs. We refer the reader to Nardelli et al. (2003), Li and Guo (2004), and Bhosle (2005) for extensive references to a variety of the MVE problem.

Most previous studies were based on the assumption that the traveller only pays attention to the difference between the lengths of the replacement path or the detour and the shortest path. In the network given by Fig. 1, the shortest path from s to t is $P_G(s, t) = (s, w, v, g, q, u, t)$ whose length is 6, the edge $e = (s, w)$ and the edge $e = (w, v)$ are the most vital edges and the edge $e = (w, v)$ is the detour critical edge of $P_G(s, t)$. If the edge $e = (w, v)$ fails, the detour is $P_{G-e}(w, t) = (w, p, u, t)$ whose length is 10, the length of the detour minus the length of $P_G(w, t) = (w, v, g, q, u, t)$ is 5 and the length of the detour is twice of the length of $P_G(w, t)$. However, if the edge $e = (u, t)$ fails, the length of the detour $P_{G-e}(u, t) = (u, p, t)$ minus the length

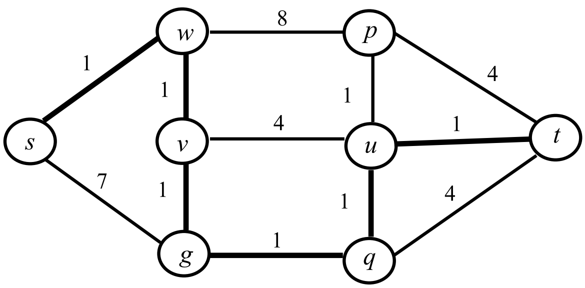

Fig. 1 MVE and the detour-critical edge analysis. In *bold*, the shortest path from s to t

of $P_G(u,t) = (u,t)$ is 4, the length of the detour is 5 times of the length of $P_G(u,t)$. From the point of view of a traveller, the edge $e = (u,t)$ is more important than the edge $e = (w,v)$.

In this paper, we define a different parameter for measuring the vitality of an edge of the shortest path $P_G(s,t)$ between the source s and destination t in $G = (V,E)$. We will face the problem of finding an edge $e = (u,v)$ in $P_G(s,t)$ whose removal produces a detour at node u such that the ratio of the length of the shortest path $P_{G-e}(u,t)$ in $G - e = (V, E - e)$ to the length of the shortest path $P_G(u,t)$ in G is maximum. We call such an edge an anti-block vital edge (AVE for short).

Our approach of building all the detours along $P_G(s,t)$ reveals its importance in network applications. Under the assumption that a sudden blockage of an edge is possible in a transportation network, a traveller may reach a node u from which he can not continue on his path as intended, just because the outgoing edge $e = (u,v)$ to be taken is currently not operational, the traveller should route along a shortest path from u to t in $G - e = (V, E - e)$. It is important to know the ratio of the length of the detour $P_{G-e}(u,t)$ to the length of $P_G(u,t)$ and the anti-block vital edge in advance, and, the maximum ratio is a key parameter for measuring the competitive ratio of a strategy for the online blockage problems such as the Canadian Traveller Problem (see Papadimitriou and Yannakakis 1989; Bar-Noy and Schieber 1991; Su 2005).

We show that the problem of finding the AVE can be solved in $O(mn)$ time, where n and m denote the number of nodes and edges in the graph, respectively. Some applications of the AVE for two special traffic networks are shown.

2 Problem statement and formulation

Let $G = (V,E)$ denote an undirected transportation network with $|V| = n$ nodes and $|E| = m$ edges, s denote the source and t the destination, $w(e)$ denote a nonnegative real weight associated to each $e \in E$. A shortest path $P_G(s,t)$ from s to t in G is defined as a path which minimizes the sum of the weights of the edges along the path from s to t and the length of $P_G(s,t)$ is denoted as $d_G(s,t)$. $P_G(u,t)$ denotes the shortest path at a node $u \in P_G(s,t) = (s, \ldots, u, v, \ldots, t)$ from u to t in G and $d_G(u,t)$ denotes the length of $P_G(u,t)$.

Definition 1 A detour at a node $u \in P_G(s,t) = (s, \ldots, u, v, \ldots, t)$ is a shortest path $P_{G-e}(u,t)$ from u to t which does not make use of (u,v), where $G - e = (V, E - e)$. $P_{G-e}(u,t)$ is the length of $P_{G-e}(u,t)$.

Definition 2 The anti-block coefficient of an edge $e = (u,v) \in P_G(s,t) = (s, \ldots, u, v, \ldots, t)$ is the ratio c_{ut} of the length of $P_{G-e}(u,t)$ to the length of $P_G(u,t)$.

Definition 3 The anti-block vital edge (AVE for short) with respect to $P_G(s,t)$ is the edge $e' = (u', v') \in P_G(s,t) = (s, \ldots, u', v', \ldots, t)$ whose anti-block coefficient is the maximum over all anti-block coefficients of edges in $P_G(s,t)$, i.e., $c_{u't} = \max_{(u,v) \in P_G(s,t)} \{c_{ut}\}$.

In order to discuss the problem, we make the following assumptions.

(1) There is only one shortest path between s and t in G.
(2) Only one edge will be blocked in the shortest path $P_G(s,t)$.
(3) G is connected even when the blocked edge is removed.

3 A property of the anti-block coefficient

$P_G(s,t) = (v_0, v_1, \dots, v_i, v_{i+1}, \dots, v_{k-1}, v_k)$ is the shortest path from s to t, where $v_0 = s$ and $v_k = t$, $P_{G-e}(v_i, v_j)$ is the shortest path from v_i to v_j which does not make use of (v_i, v_{i+1}), where $v_i, v_j \in P_G(s,t)$ and $j = i+1, i+2, \dots, k$. Figure 2 illustrates the situation.

Theorem 1 *If a path $P_G(s,t) = (v_0, v_1, \dots, v_i, v_{i+1}, \dots, v_{k-1}, v_k)$ is the shortest path from s to t in G, where $v_0 = s$ and $v_k = t$, then $c_{v_i,v_{i+1}} \geq c_{v_i,v_{i+2}} \geq \cdots \geq c_{v_i,v_k}$ for removal of $e = (v_i, v_{i+1})$.*

Proof By the definition of the anti-block coefficient of an edge, the following equality holds:

$$c_{v_i,v_j} = \frac{d_{G-e}(v_i, v_j)}{d_G(v_i, v_j)}.$$

Since $d_{G-e}(v_i, v_j) \geq d_G(v_i, v_j)$,

$$\begin{aligned} c_{v_i,v_j} &= \frac{d_{G-e}(v_i, v_j)}{d_G(v_i, v_j)} \geq \frac{d_{G-e}(v_i, v_j) + d_G(v_j, v_{j+1})}{d_G(v_i, v_j) + d_G(v_j, v_{j+1})} \\ &= \frac{d_{G-e}(v_i, v_j) + d_G(v_j, v_{j+1})}{d_G(v_i, v_{j+1})}. \end{aligned}$$

In fact, $d_{G-e}(v_i, v_j) + d_G(v_j, v_{j+1}) \geq d_{G-e}(v_j, v_{j+1})$.

Hence,

$$c_{v_i,v_j} \geq \frac{d_{G-e}(v_i, v_{j+1})}{d_G(v_i, v_{j+1})} = c_{v_i,v_{j+1}}.$$

From the above analysis, it is known that $c_{v_i,v_{i+1}} \geq c_{v_i,v_{i+2}} \geq \cdots \geq c_{v_i,v_k}$ holds. This ends the proof. □

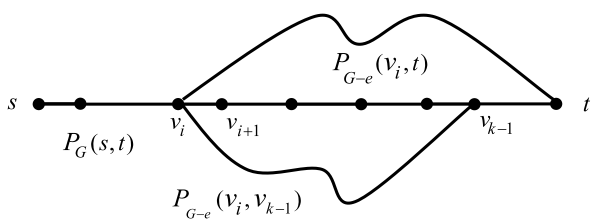

Fig. 2 Anti-block coefficient analysis in a general graph

4 Compute the anti-block vital edge

We will discuss the algorithm of finding the anti-block vital edge in a general network.

It is quite expensive to solve the AVE problem in the way of sequentially removing all the edges $e = (u, v)$ along the shortest path $P_G(s, t)$ and computing $P_{G-e}(u, t)$ at each step. In fact, this leads to a total amount of $O(n^3)$ time for the $n - 1$ edges in $P_G(s, t)$ by using the algorithm of Dijkstra (1959).

We now discuss an approach to improving the algorithm and its time complexity. We start by computing the shortest path tree rooted at t, denoted as $S_G(t)$. As shown in Fig. 3, $e = (u, v)$ denotes an edge on $P_G(s, t)$, with u closer to s than v; $M_t(u)$ denotes the set of nodes reachable in $S_G(t)$ from t without passing through edge (u, v), the length from t to the nodes in $M_t(u)$ does not change after deleting the edge e; $N_t(u) = V - M_t(u)$ denotes the set of remaining nodes (i.e., these in the subtree of $S_G(t)$ rooted at u), the distance from t to the nodes in $N_t(u)$ may increase as a consequence of deleting e. Since the detour $P_{G-e}(u, t)$ joining u and t must contain an edge in $E_t(u) = \{(x, y) \in E - (u, v) | (x \in N_t(u)) \wedge (y \in M_t(u))\}$, it follows that it corresponds to the set of edges whose weights satisfy the following condition

$$d_{G-e}(u, t) = \min_{x, y \in E_t(u)} \{d_{G-e}(u, x) + w(x, y) + d_{G-e}(y, t)\}.$$

Figure 4 illustrates the situation.

Since $S_G(t)$ is a shortest path tree rooted at t and $x \in N_t(u)$, $d_{G-e}(u, x) = d_G(u, x) = d_G(t, x) - d_G(t, u)$ and similarly, for $y \in M_t(u)$, we have $d_{G-e}(y, t) = d_G(y, t)$.

Hence, $d_{G-e}(u, t) = \min_{x \in N_t(u), y \in M_t(u)} \{d_G(t, x) - d_G(t, u) + w(x, y) + d_G(y, t)\}$.

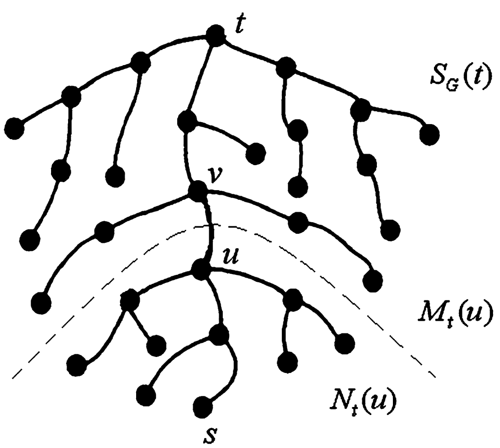

Fig. 3 $M_t(u)$ and $N_t(u)$

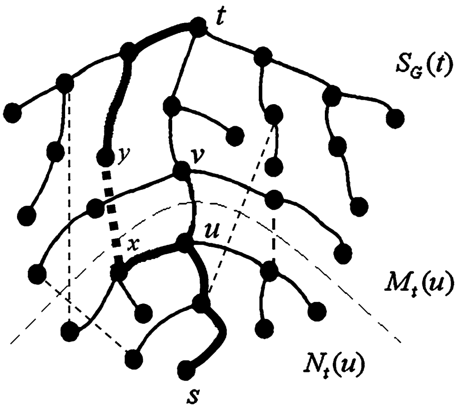

Fig. 4 Edge (u, v) is removed, *dashed lines* represent the linking edges. In *bold*, the detour at u with its linking edge (x, y)

4.1 Algorithm* for computing the anti-block vital edge

The algorithm* for obtaining the anti-block vital edge is based on the results mentioned above. Define $\{e_i = (u, v)\}, i = 1, 2, \ldots, k$ as the set of edges in $P_G(s, t)$ and $S_{G-e_i}(t)$ as the shortest path tree rooted at t which doe not make use of e_i.

Step 1. Compute the shortest path tree $S_G(t)$ rooted at t by using the algorithm of Dijkstra and record $P_G(u, t)$, $d_G(u, t)$ and k (the number of edges along $P_G(s, t)$).
Step 2. Set $i = 0$.
Step 3. Remove the edge $e_i = (u, v)$ from $P_G(s, t)$ and compute $S_{G-e_i}(t)$, $M_t(u)$ and $N_t(u)$.
Step 4. Compute $d_{G-e_i}(u, t) = \min_{x \in N_t(u), y \in M_t(u)} \{d_G(t, x) - d_G(t, u) + w(x, y) + d_G(y, t)\}$ and $c_{ut} = \frac{d_{G-e_i}(u,t)}{d_G(u,t)}$.
Step 5. Set $i = i + 1$. If $i < k$, then turn to step 3; otherwise turn to step 6.
Step 6. Compute c_{ut}, $c_{u't} = \max\{c_{ut}\}$ and the anti-block vital edge e'.

4.2 Analysis of the time complexity on the algorithm*

For the shortest path tree $S_G(t)$ can be computed in $O(n^2)$ time by using the algorithm of Dijkstra, the computation time of $S_{G-e_i}(t)$, $M_t(u)$ and $N_t(u)$ for each e_i is $O(1)$, and the computation time of step 4 is $O(m)$. Since steps 2–5 are loop computation and its repeat times is k, then the total time for steps 2–5 is $O(mn)$ for $k \leq n - 1$. The computation time of step 6 is $O(n)$. It is known that the time complexity of the algorithm* is $O(mn)$.

From the above analysis, the following theorem holds.

Theorem 2 *In a graph with n nodes and m edges, the anti-block vital edge problem on a shortest path between two nodes s and t can be solved in $O(mn)$ time.*

5 Applications of the anti-block vital edge in urban traffic networks

The grid-type network and the circular-type network are two examples of urban traffic networks. In these two special cases, the anti-block vital edges have some properties. We will discuss them in details.

As shown in Fig. 5, let $G=(V,E)$ denote an undirected planar grid-type network with $(p+1)(q+1)$ nodes, there are $p+1$ rows of nodes in horizontal direction and $q+1$ columns of nodes in vertical direction. $E=\{(v_{ij},v_{i,j+1})\}\cup\{(v_{ij},v_{i+1,j})\}$, $i=0,1,2,\dots,p$, $j=0,1,2,\dots,q$ denotes the set of the edges in G. The weight of each edge is 1. A line of G is consist of the set of nodes $\{v_{ij}|j=0,1,2,\dots,q\}$ or $\{v_{ij}|i=0,1,2,\dots,p\}$.

Theorem 3 *In an undirected planar grid-type network, if a source node and a destination node locate in a line, then the edge adjacent to destination node is the anti-block vital edge of the shortest path between the two nodes.*

Proof Let v_{00} denote a source node and v_{0q} a destination node. If there is no any blockage happening in G, the shortest path from v_{00} to v_{0q} is $P_G(v_{00},v_{0q})=(v_{00},v_{01},\dots,v_{0j},\dots,v_{0q})$.

Since $c_{v_{0j},v_{0q}}=\frac{d_{G-e}(v_{0j},v_{0q})}{d_G(v_{0j},v_{0q})}=\frac{q-j+2}{q-j}$, $c_{v_{0,j+1},v_{0q}}=\frac{d_{G-e}(v_{0,j+1},v_{0q})}{d_G(v_{0,j+1},v_{0q})}=\frac{q-j-1+2}{q-j-1}$ and $\frac{q-j+1}{q-j-1}>\frac{q-j+2}{q-j}$, then $c_{v_{0j},v_{0q}}<c_{v_{0,j+1},v_{0q}}$.

Similarly, the following inequality holds: $c_{v_{00},v_{0q}}<c_{v_{01},v_{0q}}<\cdots<c_{v_{0,q-1},v_{0q}}$.

It is known that the anti-block vital edge is the edge adjacent destination node in the shortest path under the assumption of source node and destination node locating on a line in a grid-type network. This ends the proof. □

As shown in Fig. 6, $G=(V,E)$ denotes an undirected planar circular-type network with $(p+1)(q+1)+1$ nodes, $E=\{(s,v_{0j})\}\cup\{(v_{ij},v_{i,j+1})\}\cup\{(v_{ij},v_{i+1,j})\}$, $i=0,1,2,\dots,p$, $j=0,1,2,\dots,q$ denotes the set of the edges in G. Let the polar

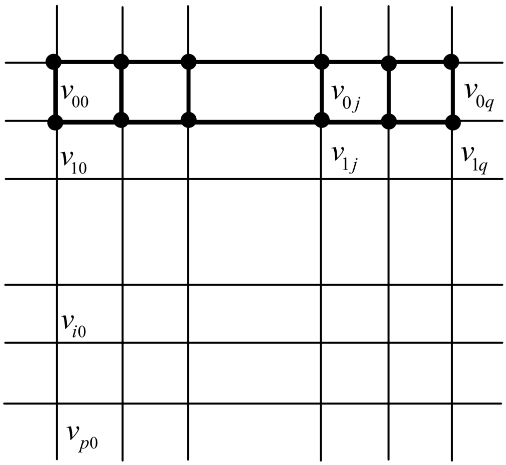

Fig. 5 Anti-block coefficient analysis in a grid-type network

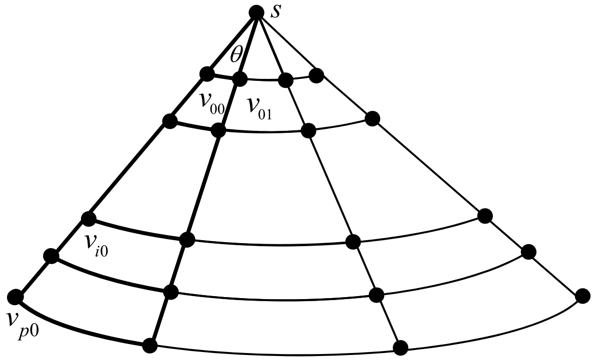

Fig. 6 Anti-block coefficient analysis in a circular-type network

angle θ of the edge $e = (s, v_{00})$ and the edge $e = (s, v_{01})$ satisfy $\theta < \frac{\pi}{2}$, the weight of edge and $e = (s, v_{0j})$ $e = (v_{ij}, v_{i+1,j})$ be 1. Therefore, the weight of $e = (v_{i0}, v_{i1})$ is $(i+1)\theta$, $i = 0, 1, 2, \ldots, p$.

Theorem 4 *In an undirected planar circular-type network, if v_{p0} is a source node and s is a destination node, then every edge e of the shortest path $P_G(v_{p0}, s)$ from v_{p0} to s is an anti-block vital edge.*

Proof If there is no any blockage happening in G, the shortest path from v_{p0} to s is $P_G(v_{p0}, s) = (v_{p0}, \ldots, v_{i0}, \ldots, v_{10}, v_{00}, s)$.

Since $c_{v_{i0},s} = \frac{d_{G-e}(v_{i0},s)}{d_G(v_{i0},s)} = \frac{(i+1)\cdot\theta+(i+1)}{i+1} = \theta + 1$, $i = 0, 1, 2, \ldots, p$, then the following equality holds: $c_{v_{p0},s} = c_{v_{p-1,0},s} = \cdots = c_{v_{i0},s} = \cdots = c_{v_{00},s} = \theta + 1$. It is known that every edge along the shortest path $P_G(v_{p0}, s)$ is the anti-block vital edge of $P_G(v_{p0}, s)$. This ends the proof. □

6 Conclusions

From the transportation network management point of view, it is valuable to identify the result by the failure of a component. In this paper, under the assumption that a sudden blockage of an edge is possible in a transportation network, we define a different parameter—anti-block vital edge (AVE for short) for measuring the vitality of an edge along the shortest path $P_G(s, t)$ between the source s and destination t in G. Our approach of building all the detours along $P_G(s, t)$ and the ratio of the length of the detour $P_{G-e}(u, t)$ to the length of $P_G(u, t)$ for each edge $e = (u, v) \in P_G(s, t)$ reveals its importance in network applications. The maximum ratio is also a key parameter for measuring the competitive ratio of a strategy for the online blockage problems. We show that the problem of finding the AVE can be solved in $O(mn)$ time in a general network, where n and m denote the number of nodes and edges in the graph, respectively. Some applications of the AVE for two special traffic networks are shown. There are some further directions of future work, including improving the present algorithm for computing the AVE and developing algorithm for finding the anti-block vital node when blockage happens at a node.

References

Bar-Noy A, Schieber B (1991) The canadian traveler problem. In: Proceedings of the second annual ACM-SIAM symposium on discrete algorithms, pp 261–270

Bhosle AM (2005) Improved algorithms for replacement paths problems in restricted graphs. Oper Res Lett 33:459–466

Corley HW, Sha DY (1982) Most vital links and nodes in weighted networks. Oper Res Lett 1:157–161

Dijkstra EW (1959) A note on two problems in connexion with graphs. Numer Math 1:269–271

Hershberger J, Suri S (2001) Vickrey prices and shortest paths: What is an edge worth? In: Proceedings of the 42nd annual IEEE symposium on foundations of computer science, pp 252–259

Hershberger J, Suri S, Bhosle A (2003) On the difficulty of some shortest path problems. In: Proceedings of the 20th symposium on theoretical aspects of computer science, pp 343–354

Li Y, Guo Y (2004) Study on vital edges of shortest paths in traffic and transportation networks. Chin J Manag Sci 12(4):69–73

Malik K, Mittal AK, Gupta SK (1989) The *k* most vital arcs in the shortest path problem. Oper Res Lett 8:223–227

Nardelli E, Proietti G, Widmayer P (1998) Finding the detour critical edge of a shortest path between two nodes. Inf Process Lett 67(1):51–54

Nardelli E, Proietti G, Widmayer P (2001) A faster computation of the most vital edge of a shortest path between two nodes. Inf Process Lett 79(2):81–85

Nardelli E, Proietti G, Widmayer P (2003) Finding the most vital node of a shortest path. Theor Comput Sci 296:167–177

Papadimitriou CH, Yannakakis M (1989) Shortest paths without a map. In: Ronchi Della Rocca S, Ausiello G, Dezani-Ciancaglini M (eds) Automata, languages and programming. Lecture notes in computer science, vol 372. Springer, Heidelberg, pp 610–620

Su B (2005) Research on strategy for sequential unexpected blockages during the transportation process. PhD dissertation