Research on an Algorithm of the Acceptable Shortest Path
Dong-dong HE*, Yin-zhen LI, Jia-jie SHEN and Wen LI
School of Traffic and Transportation, Lanzhou Jiaotong University, Lanzhou 730070, China
*Corresponding author

Keywords: The acceptable shortest path, Traffic and transportation network, Vital edge, Algorithm complexity.

Abstract. In transportation network, the path interruption occurs frequently, and how to choose a reasonable path before departure is the key to ensure the timely delivery of materials. In this paper, on the basis of defining the alternative acceptable coefficient of the edge, the alternative acceptable coefficient of the path and the acceptable shortest path (ASP), the algorithm of the ASP is proposed, which can guarantee the transportation time is still acceptable after path interruption, and its complexity is also analyzed. Finally, a numerical example is performed to validate the efficiency and rationality of the proposed algorithm, and the results indicate that the proposed algorithm provides a good way to solve such kinds of problems in the fields of traffic, communication and so on.

Introduction

The problem of the most vital edge in shortest path is firstly proposed by Corley and Sha in 1982, in a given 2-edge connected network $G = (V, E)$ with $|V| = n$ vertexes and $|E| = m$ edges, then there are at least one edge $e^*$ whose removal from $G = (V, E)$ will result the greatest increase in the shortest distance between two specified nodes called the origin node $s$ and the destination node $t$, such an edge $e^*$ is called the most vital edge of the shortest path [1]. In 1984, Fredman and Tarjan used the Fibonacci Heap to improve network optimization [2]; Malik et al. proposed an algorithm of vital arc by using the Fibonacci Heap in 1989 which time complexity is $O(m + n \log n)$ [3]. Nardelli et al. gave an algorithm of finding the detour-critical edge of a shortest path between two nodes with the time complexity $O(m + n \log n)$ in 1998 [4], then they optimized the time complexity to $O(m \cdot \alpha(m, n))$, where $\alpha$ is the functional inverse of the Ackermann function [5]; In 2004, Li Yinzhen and Guo Yaohuang gave a sub-tree connectivity algorithm to find the vital edge of the shortest path with time complexity $O(n^2)$ [6]; In 2006, Yan Huahai and Xu Yinfeng presented an algorithm about the most vital edge of the shortest path problem under incomplete information [7]. In 2008, Su Bing and Xu Qingchuan proposed a new concept—the anti-block vital edge problem with the algorithm complexity $O(mn)$ [8]. After that, Nie Zhe and Li Yueping optimized the time complexity to $O(m + n \log n)$ as well, and mentioned that if they use the inverse function of Ackerman that the algorithm complexity can also be reduced to $O(m \cdot \alpha(m, n))$ [9]. In 2009, Xiao Peng and Xu Yinfeng et al. proposed the anti-risk path problem between two nodes in undirected graphs with time complexity $O(mn + n^2 \log n)$ [10]. By research, the algorithm complexity in the literatures mentioned above which less than $O(n^2)$ are basically used the computer data access or the definition of data structures. The vital edge plays an important role in practical applications, once it breaks down, it will influence some paths or even the whole network.

In reality, such a situation often exists: the traveler chose a path before departure, when arrived at a certain node, we find the path to the next node was interrupted by abrupt events, so that we will no longer follow the planned route, generally, we will not consider to return to the starting point and then re-select a new path, but detour to the destination node from the node that we have reached. However,
it will generate the extra travel time which may be not acceptable for the deadline. Thus it is necessary to solve such a problem: how to choose the ASP before departure to address the abrupt disruptions of the planned route which can also ensure that the transit period is still acceptable.

Suppose that a shipment of emergency supplies need to be transported from \( s \) to \( t \), the acceptable delivery time is 18 days. Hypothesis that the delivery time of the 1st shortest path from \( s \) to \( t \) is 10 days, if the most vital edge in the first shortest path is broken, then the transport time will be increased to 20 days. On the other hand, if we choose the 2nd shortest path and its transport time is 13 days, once the most vital edge of the path is interrupted, the delivery time will increase to 17 days. Taking the worst case into account, the path we can accept is more likely the 2nd shortest path. Therefore, this paper discussed and studied the related problems of the ASP based on similar problems mentioned above.

**Problem Description and Definition**

Let \( G=(V,E) \) be a connected network, there must be the 1st shortest path, the 2nd shortest path, ..., the \( k \)th shortest path from node \( s \) to node \( t \), denoted as \( P_1(s,t), P_2(s,t),..., P_k(s,t) \), for convenience, we denote as \( P_1, P_2,..., P_k \) for short, their corresponding length of the paths denote as \( D_{P_1}, D_{P_2},..., D_{P_k} \); their set denote as \( P = \{ P_i | P_i \) is the \( i \)th shortest path in \( G \), \( i = 1,2,...,k \} \).

**Lemma 1**[1]. In the connected network \( G=(V,E) \), and all of \( P_1, P_2,..., P_{k-1} \) contain edge \( e=(u,v) \), that is \( e \in \{ P_1 \cap P_2 \cap ... \cap P_{k-1} \} \), so the \( P_{k} \) is the first shortest path which does not contain \( e \) in \( G=(V,E-e) \).

**Proof. Proof by contradiction.** Suppose that \( P_k \) is not the first shortest path in network \( G=(V,E-e) \) and denote the length of \( P_k \) by \( D_k \), so there exists a path \( P_{k'} \) in \( G=(V,E-e) \) and the corresponding length is \( D_{k'} \), then we have \( D_k > D_{k'} \), at the same time, \( P_{k'} \) is one of the shortest paths in \( G=(V,E) \) as well and shorter than \( P_k \), so \( P_{k'} \) should be the \( k \)th shortest path in \( G=(V,E) \), contradict with the subject hypothesis. This completes the proof.

In \( G=(V,E) \), the \( P_i(s,t) \) is the \( i \)th shortest path from node \( s \) to node \( t \), when the edge \( e=(u,v) \) of the path \( P_i(s,t) \) is interrupted, the length of the shortest path from \( u \) detour to \( t \) denote as \( D_{P_i-e}(u,t) \), so the length of the shortest path from \( s \) to \( u \) and \( u \) detour to \( t \) is denoted as \( D_{P_i-e}(s,t) \).

**Definition 1.** The alternative acceptance coefficient of the edge \( e_j \) (where \( e_j \) is the \( j \)th edge of path \( P_i(s,t) \) ) is \( \delta_j = \frac{D_{P_i-e_j}(s,t) - D_{P_i}(s,t)}{D_{P_i}(s,t)} \); The alternative acceptability of path \( P_i(s,t) \) is \( \delta_P = \max \{ \delta_j | j=1,2,...,l \} \), (where \( l \) is determined by the number of edges of path \( P_i(s,t) \)).

**Definition 2.** For a given path acceptability \( \delta \), if exists \( \delta_P < \delta \), the path \( P_i(s,t) \) is called the ASP.

For the sake of convenience and without loss of generality, we suppose that:

1. For paths \( P_1, P_2,..., P_k \), the corresponding lengths, separately, are \( D_{P_1}, D_{P_2},..., D_{P_k} \) and we have \( D_1 \leq D_2 \leq ... \leq D_k \).
2. Only one interruption will be happen on a given path.
3. When any edge \( e \) in \( G=(V,E) \) is interrupted, \( G=(V,E-e) \) is still connected.

The square network and sector (or circular) network are typical networks in urban road system, such as the road networks in Xi’an and Chengdu which can, separately, be seen as the square and circular network. Therefore, it is necessary to study the ASP problem in these two classical networks, additionally, two simple theorems are proposed here.

**Theorem 1.** In a planar square network \( G=(V,E) \), if the length (weight) between two adjacent
nodes is 1. If the path $P_i(s,t)$ (where $s$ and $t$ locate in a same line) is the ASP, then its sufficient condition is:

$$D_h(s,t) \geq \frac{2}{\delta}$$

where $\delta$ is the given path acceptability, $D_h(s,t)$ is the length (weight) of path $P_i(s,t)$.

**Proof.** Let the positions of the source node $v_{ip}(s)$ and the sink node $v_{iq}(t)$ as shown in Fig. 1, the first shortest path from node $s$ to node $t$ is $P_i(s,t) = \{v_{ip}, v_{ip(p+1)}, \ldots, v_{ip(q-1)}, v_{iq}\}$ (where $s = v_{ip}, t = v_{iq}$), and its length is $D_h(s,t) = q - p$, it is easy to know that if the $jth$ edge of the $P_i(s,t)$ is broken, its corresponding detour distance is 2 units more than the first shortest path, that is $D_{h-e_j}(s,t) = q - p + 2$, as shown in Figure 1, hence,

$$\delta_n = \max \{\delta_j \mid j \in J\} = \max \left\{ \frac{D_{h-e_j}(s,t) - D_h(s,t)}{D_h(s,t)} \mid j \in J \right\} = \frac{q - p + 2 - (q - p)}{q - p} = \frac{2}{q - p} \leq \frac{2}{D_h(s,t)}$$

where $J = \{1, 2, \ldots, (q - p)\}$.

For the given path acceptability $\delta$, if the first shortest path from the node $s$ to the node $t$ is the ASP that $\delta_n = \frac{2}{D_h(s,t)} \leq \delta$ must be established, that is $D_h(s,t) \geq \frac{2}{\delta}$. This ends the proof.

\[ \text{Figure 1. A brief analysis of square networks.} \]

\[ \text{Figure 2. A brief analysis of sector network.} \]

**Theorem 2.** In the planar sector networks with every polar angle is $\theta$ ($\theta < \frac{\pi}{2}$) respectively, the length (weight) between two adjacent nodes on the same radius is 1, and let the node $s$ and node $t$ are in the same radius, $t$ is the center of the circle. If the path $P_i(s,t)$ (where $s$ and $t$ locate in a same radius) is the ASP, then its sufficient condition is:

$$\theta \leq \delta$$

where $\delta$ is the given path acceptability.

**Proof.** Suppose that the positions of the source node $v_{ip}(s)$ and the destination node $t$ are shown
as in Fig 2. The first shortest path between $s$ and $t$ is $P_1(s,t) = \{v_0, v_{i(p+1)}, ..., v_{i(p+q-1)}, v_{i(p+q)}, t\}$, (where $s = v_0$), its length is $D_{\overline{p}}(s,t) = q + 1$, it is easy to know that if the $j$th edge of the $P_1(s,t)$ is broken, its corresponding detour distance is $r(j)\theta$ units more than the first shortest path, that is $D_{\overline{p}-e_j}(s,t) = q + 1 + r(j)\theta$. The $r(j)$ is a function of the edge $e_j$, which depends on different broken edges.

The general formula of $D_{\overline{p}-e_j}(s,t)$ is easy to get:

$$D_{\overline{p}-e_j}(s,t) = q + 1 + r(j)\theta$$

so

$$\delta_{\overline{p}} = \max \{\delta_{e_j} \mid j \in J\}$$

$$= \max \left\{ \frac{D_{\overline{p}-e_j}(s,t) - D_{\overline{p}}(s,t)}{D_{\overline{p}}(s,t)} \mid j \in J \right\}$$

$$= \max \left\{ \frac{q + 1 + (q + 2 - j)\theta - (q + 1)}{q + 1} \mid j \in J \right\}$$

$$= \max \left\{ \frac{\theta - \frac{j - 1}{q + 1}\theta}{j \in J} \right\}$$

where $J = \{1, 2, ..., q, (q + 1)\}$.

Because $q, \theta$ are constant values and $j \in J = \{1, 2, ..., q, (q + 1)\}$ is a gradually increasing positive integer; So, when $j = 1$, $\delta_{\overline{p}}$ gets the maximum value. For the given path acceptability $\delta$, if the 1st shortest path $P_1(s,t)$ between the node $s$ and the node $t$ is the ASP, the following equality holds: $\delta_{\overline{p}} = \delta_1 = \theta \leq \delta$, that is $\theta \leq \delta$ established. This completes the proof.

Find the ASP

Now let’s study the algorithm of the ASP in a general network $G = (V, E)$, the shortest path spanning tree $S_{\overline{p}}(t)$ with the root $t$ (where $t$ is the destination node) can be computed by references [3][4]. Let $e = (u, v)$ is one of edge in $P_\sigma(s,t)$, $e \in P_\sigma(s,t)$, where the node $u$ is closer to $s$ than $t$. Let $M_i(u)$ be a set of nodes in $S_{\overline{p}}(t)$ whose nodes can be reachable directly from the destination node $t$ without passing through the edge $e$, at the same time, the rest of nodes in the $S_{\overline{p}}(t)$ are denoted as $N_i(u) = V - M_i(u)$. $N_i(u)$ can be considered as the shortest path spanning sub-tree with the root $u$. Fig 3 illustrates the situation.

It is found that the distance between the nodes in $M_i(u)$ and node $t$ has not changed when the edge $e$ is interrupted, but the distance from the nodes in $N_i(u)$ to node $t$ has changed. On the other hand, dividing the nodes set $V$ into $M_i(u)$ and $N_i(u)$ is equivalent to define a cut in $G - e$, and then there exists an edge set $E_i(u)$:

$$E_i(u) = \{(x, y) \in E - (u, v) \mid x \in N_i(u) \text{and} y \in M_i(u)\}$$
Due to the interruption of $e = (u, v)$, the detour path $P_{G-e}(u, t)$ from $u$ to $t$ must contains one of edge in $E_e(u)$ and the following equality holds(see Figure 4.):

$$
D_{G-e}(u, t) = \min_{(x, y) \in E_e(u)} \{D_{G-e}(u, x) + w(x, y) + D_{G-e}(y, t)\}
$$

(1)

Since $x \in N_t(u)$, we obtain:

$$
D_{G-e}(u, x) = D_G(u, x) = D_G(t, x) - D_G(t, u)
$$

(2)

Similarly $y \in M_t(u)$, we obtain:

$$
D_{G-e}(y, t) = D_G(y, t)
$$

(3)

From above equalities (1)(2)(3), we obtain:

$$
D_{G-e}(s, t) = D_{G-e}(s, u) + D_{G-e}(u, t)
$$

$$
= D_G(s, u) + \min_{(x, y) \in E_e(u)} \{D_G(t, x) - D_G(t, u) + w(x, y) + D_G(y, t)\}
$$

(4)

Algorithm Steps of the ASP

Step 1. Using the Bellman-Ford[11] algorithm to find the 1st, 2nd, ..., kth shortest path, denote as $P_1, P_2, \ldots, P_k$ and they satisfy the equality $\frac{D_{P_1} - D_{P_k}}{D_{P_1}} < \delta$, that is $D_{P_1} < D_{P_1}(1 + \delta)$. Their collections are recorded as $P = \{P_i | i = 1, 2, \ldots, k\}$, their corresponding lengths are denoted as $D_{P_1}, D_{P_2}, \ldots, D_{P_k}$ and we have $D_{P_1} < D_{P_2} < \ldots < D_{P_k}$. Let $D_{P_1} = D$, $E_P = \{e_{i_1}, e_{i_2}, \ldots, e_{i_\delta}\}$ representing the edges’ order of path $P_i$ (the number of edges $l$ is different for different paths);

Step 2. Let $i = 1$;

Step 3. Let $j = 1$;

Step 4. Let $\delta_i = 0$;

Step 5. Delete $e_j$, calculate $S_{G-e_j}(t)$, $M_{P_i}(u)$, $N_{P_i}^P(u)$.

Step 6. Calculate:
Suppose that Figure 5. is a road transport network in a certain area where is a mountainous region with
continuous heavy rainfall from June to September each year, which makes landslide and debris flow
occur frequently and causes the traffic disruption and serious economic losses; Therefore, it is very
significant to choose an ASP under the uncertainty road conditions.

Firstly, for the assumed path acceptability \( \delta = 0.8 \), using the algorithm in this paper we have
\( P_1 = (s-a-b-t) \), \( D_{R_1} = 10 \); \( P_2 = (s-a-c-b-t) \), \( D_{R_2} = 12 \); \( P_3 = (s-g-h-t) \), \( D_{R_3} = 14 \);
\( P_4 = (s-g-h-e-t) \), \( D_{R_4} = 16 \); \( P_5 = (s-g-f-h-t) \), \( D_{R_5} = 18 \); but \( P_5 \) does not satisfy
\( D_{R_5} < D_{R_1}(1 + \delta) \); So there are only 4 alternative paths, that is \( P = \{ P_1, P_2, P_3, P_4 \} \), let \( D_{R_i} = D = 10 \),
\( E_{R_i} = \{ e_1, e_2, \ldots, e_d \} \).

Then delete \( e_{i_1} \) (see Fig. 6), we get \( S_{G-e_{i_1}}^R(t) \), \( M_i^R(u) = \{ a, b, c, d, e, f, g, h, t \} \), \( N_i^R(u) = \{ s \} \). Calculate
\( D_{R-e_{i_1}}(s,t) = 14 \), \( \delta_{i_1} = 0.4 \), \( \delta_{R_1} = 0.4 \). Then delete other edges of \( E_{R_i} \) in turn, calculate \( D_{R-e_{i}}(s,t) \), \( \delta_{i} \)
and obtain \( \delta_R = \max\{0.4,0.2,1.0\} = 1.0 \), but it does not satisfy \( \delta_R \leq \delta \).
Similarly, we have \( \delta_1 = \max\{0.4, 0, 0.8, 1.2\} = 1.2 \), but it does not satisfy \( \delta_1 \leq \delta \), either; \( \delta_2 = \max\{0, 0.8, 0.6\} = 0.8 \), we have \( \delta_2 \leq \delta \), the algorithm is stopped and the calculation results are shown in Table 1.

<table>
<thead>
<tr>
<th>( p )</th>
<th>( D_s )</th>
<th>( \delta_1 )</th>
<th>( \delta_2 )</th>
<th>( \delta_3 )</th>
<th>( \delta_4 )</th>
<th>( \delta_5 )</th>
<th>( \delta_2 \leq \delta )?</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_1 )</td>
<td>10</td>
<td>0.4</td>
<td>0.2</td>
<td>1.0</td>
<td>—</td>
<td>1.0</td>
<td>No</td>
</tr>
<tr>
<td>( p_2 )</td>
<td>12</td>
<td>0.4</td>
<td>0</td>
<td>0.8</td>
<td>1.2</td>
<td>1.2</td>
<td>No</td>
</tr>
<tr>
<td>( p_3 )</td>
<td>14</td>
<td>0</td>
<td>0.8</td>
<td>0.6</td>
<td>—</td>
<td>0.8</td>
<td>Yes!(stop)</td>
</tr>
<tr>
<td>( p_4 )</td>
<td>16</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Therefore, the ASP is \( P_S = (s - g - h - t) \) between \( s \) and \( t \), it can be interpreted as: for the assumed path acceptability \( \delta = 0.8 \), even if one of the section \( s - g \) or \( g - h \) or \( h - t \) on \( P_S \) interrupted, its negative impact on the arrival time is still controllable, this path can also guarantee that the period of shipment is still acceptable.

Conclusions

In the transportation network, the road interruption happens frequently due to all kinds of unexpected events which makes it difficult to follow the planned transportation path, it has a great negative impact on the arrival time as well. So it is necessary to study the ASP, we put forward the algorithm running in \( O(n^3) \) time which is an efficient algorithm. In the future research, the following questions need to be further considered: (1) If multiple edges simultaneous interrupts with a certain probability, how to choose the ASP? (2) How to define efficient data structures to reduce the complexity of this algorithm and speed up the search for the ASP. (3) How to obtain the ASP under terminal uncertainty[12].

Acknowledgements

The authors would like to thank the anonymous referees for their suggestions, which helped us to improve the paper.

References


