Hindawi
Journal of Advanced Transportation
Volume 2017, Article ID 9216864, 16 pages
https://doi.org/10.1155/2017/9216864

WILEY | Hindawi

Research Article

Three Extensions of Tong and Richardson's Algorithm for Finding the Optimal Path in Schedule-Based Railway Networks

J. Xie,[1] S. C. Wong,[1] and S. M. Lo[2]

[1]*Department of Civil Engineering, The University of Hong Kong, Pokfulam, Hong Kong*
[2]*Department of Architecture and Civil Engineering, City University of Hong Kong, Kowloon, Hong Kong*

Correspondence should be addressed to J. Xie; jiemin@hku.hk

Received 2 July 2016; Accepted 26 October 2016; Published 12 January 2017

Academic Editor: Richard S. Tay

High-speed railways have been developing quickly in recent years and have become a main travel mode between cities in many countries, especially China. Studying passengers' travel choices on high-speed railway networks can aid the design of efficient operations and schedule plans. The Tong and Richardson algorithm that is used in this model offers a promising method for finding the optimal path in a schedule-based transit network. However, three aspects of this algorithm limit its application to high-speed railway networks. First, these networks have more complicated common line problems than other transit networks. Without a proper treatment, the optimal paths cannot be found. Second, nonadditive fares are important factors in considering travel choices. Incorporating these factors increases the searching time; improvement in this area is desirable. Third, as high-speed railways have low-frequency running patterns, their passengers may prefer to wait at home or at the office instead of at the station. Thus, consideration of a waiting penalty is needed. This paper suggests three extensions to improve the treatments of these three aspects, and three examples are presented to illustrate the applications of these extensions. The improved algorithm can also be used for other transit systems.

1. Introduction

Traffic assignment in a transit network is unlike that in a road network, because transit vehicles run on fixed routes and predetermined timetables. Vehicles on roads have more freedom in route selection, and they do not need to depart at specific times [1]. There are two approaches to consider the time dimension and the fixed routes in a transit network, that is, the frequency-based model and the schedule-based model. The frequency-based model [2–4] simplifies the transit network by regarding each line as having many runs and the headway of runs as having a mean. The schedule-based model [1] is based on the real timetable. Thus, in evaluating the characteristics of these two approaches, it seems that the scheduled-based model is more promising for dynamic transit networks. Generally, there are three methodologies for path generation in a schedule-based network.

1.1. Branch and Bound Method. Tong and Richardson [1] suggested an optimal path algorithm which combined Dijkstra's algorithm [5] with the branch and bound method. Later, Tong and Wong [6] extended this algorithm by considering nonadditive fares. Khani et al. [7] improved the searching speed of this algorithm by setting a lower bound to reduce the size of searches.

Other researchers have introduced several important variables to limit path searching. These variables include the latest departure time [8], the preset time window between the planned departure time and the arrival time [9–14], the maximum number of transfers [8, 9, 11, 13], the maximum waiting time for transfers [9, 11], and the walking distance [13]. Applying these variables (or bounds) can improve the branch and bound method, but the effectiveness of the bounds depends on their preset values. Compared with Tong and Richardson's algorithm, these bounds are not tight enough to effectively reduce the searching size.

1.2. Event Dominance Method. Florian [15] suggested an event dominance method, which can be used to find optimal paths to one or multiple destinations. However, the optimality

of the path found by this method is questionable, because the searching stops after one event associated with the destination is found. In addition, the real optimal path might be ruled out at an earlier stage. This problem was investigated by Nielsen and Frederiksen [16] using a counterexample. They reported using hidden waiting time at the origin point to solve it and explained the method with a two-line example. However, the search size then becomes larger, so they needed to also use the graph cut approach, a modification that may fail to help if the optimal path has more than one transfer, due to the event dominance principle.

1.3. Other Methods. Nielsen et al. [17] suggested a specified design method for the Copenhagen suburban rail network. They divided the path choices into two kinds, that is, paths with no transfers and paths with transfers. The optimal path with no transfers was selected as the first train departing at the origin to the destination after the planned departure time. For the path with transfers, the whole path was divided into several segments, according to the transfer points that were planned in advance, based on the network. Each of the segments is treated like the path with no transfer. This method greatly reduces the computation time, but it is only suitable for simple networks with several transfer stations and with full information about passenger choices.

Chen and Yang [18] modified the traditional graphic network by adding departure time windows on nodes. That is, if the arrival time at node i is not within the preset departure time window, the passengers will need to wait and decide whether to depart at the next time window. This kind of algorithm provided a way to consider the time dimension, but it is not a real schedule-base algorithm, and it simply treats waiting time in the same way as the in-vehicle time.

Compared with the other approaches, Tong and Richardson's algorithm has an improved capacity for effectively and flexibly handling complex transit networks in real time. However, in its present form it still has three limitations. This paper aims to extend Tong and Richardson's algorithm to improve its results in dynamic transit networks. This improved algorithm can be used as a tool for information provision. Passengers can plan the best path according to their own preference, such as one without transfer or with less waiting time. In the future, we will develop this algorithm for network assignment and the planning of schedules, but at this stage the algorithm is mainly for transit trip navigation.

2. Tong and Richardson's Algorithm

2.1. Description of Tong and Richardson's Algorithm. The Tong and Richardson algorithm can be described as follows.

Step 1. The time-dependent quickest path algorithm is developed from Dijkstra's algorithm [5], and it uses travel time between nodes as the path cost (as based on the timetable) to search for the time-dependent quickest paths from the origin o to all other nodes, if departing at T_o^{DE}. The variable T_o^{DE} is the actual clock time for departure at the origin o, unit: mins. After the quickest path search, the earliest arrival time at node i, namely, T_i^{EA}, can be found.

Step 2. Based on the quickest path Q from the origin o to the destination d, the total weighted cost W_{od}^{Q} can be calculated. The weighted cost function for path j from node i to node q is

$$W_{iq}^{j} = f\left(t_{iq}^{j}\right) + S_{iq}^{j} * P^{\mathrm{T}}, \tag{1}$$

where t_{iq}^{j} is the journey time (including waiting time, walking time, and transit time) for path j from node i to q, unit: mins.

$f(t_{iq}^{j})$ is the weighted function for journey time in using path j, unit: mins:

$$f\left(t_{iq}^{j}\right) = P^{\mathrm{WK}} * t_{j}^{\mathrm{WK}} + P^{\mathrm{W}} * t_{j}^{\mathrm{W}} + P^{\mathrm{IV}} * t_{j}^{\mathrm{IV}}, \tag{2}$$

where

$$t_{iq}^{j} = t_{j}^{\mathrm{WK}} + t_{j}^{\mathrm{W}} + t_{j}^{\mathrm{IV}}, \tag{3}$$

where t_{j}^{WK} is the walking time for path j, unit: mins; t_{j}^{W} is the waiting time at stations for path j, unit: mins; t_{j}^{IV} is the in-vehicle time for path j, unit: mins; P^{WK} is the penalty for walking time; P^{W} is the penalty for waiting time; P^{IV} is the penalty for in-vehicle time; P^{T} is the penalty for transfer, unit: mins; S_{iq}^{j} is the number of transfers for path j from node i to q.

Step 3. Set the upper boundary of the latest arrival time at the destination d:

$$B_1 = W_{od}^{\mathrm{Q}} + T_o^{\mathrm{DE}}. \tag{4}$$

Step 4. Reverse the timetable, and set the reversed upper boundary as the departure time for searching the time-dependent quickest paths from the destination to all other nodes, so that we have the earliest arrival time at node i from the destination d in the reversed network T_i^{EAR}. Then, reverse T_i^{EAR} to get the latest departure time to the destination d at node i, T_i^{LD}.

Step 5. Compare T_i^{EA} and T_i^{LD} to check the accessibility of node i, A_i:

$$A_i = \begin{cases} 1, & \text{if } T_i^{\mathrm{LD}} - T_i^{\mathrm{EA}} \geq 0, \text{ node } i \text{ is accessible,} \\ 0, & \text{if } T_i^{\mathrm{LD}} - T_i^{\mathrm{EA}} < 0, \text{ node } i \text{ is inaccessible.} \end{cases} \tag{5}$$

Exclude inaccessible nodes, so that inaccessible nodes are not included in the optimal path search (i.e., in Step 6).

Step 6. The optimal path search finds all of the possible paths by searching all accessible nodes. The weighted time function W_{oi}^{j} is calculated for node i along path j. If $W_{oi}^{j} > W_{od}^{\mathrm{Q}}$, the search for path j ends.

Step 7. The smallest $W_{od}^{j^*}$ is found, and the path j^* is identified as the optimal path from the origin o to destination d.

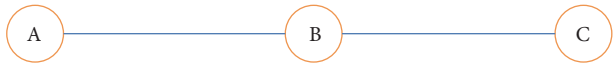

Figure 1: An example network for the common lines problem.

2.2. Limitations of the Existing Tong and Richardson Algorithm. This algorithm can be used for most transit networks. However, it cannot deal with the following three common situations in railway networks.

2.2.1. Nonadditive Fare. Tong and Richardson's algorithm does not consider travel fares. If the fare is link-based and additive (i.e., the fare proportionally increases with the distance or travel time), Tong and Richardson's algorithm still works when adding an additional linear extension for the weighted time function (such as that provided by Friedrich et al. [9]) along the optimal path search. However, the fare system for the railway is usually nonadditive, so the additive-fare function [9] cannot be used. A later improvement by Tong and Wong [6] allowed for the consideration of the non-additive fare. However, introducing the fare into the weighted cost function enlarged the bounding (i.e., the searching size), so Tong and Wong used the cheapest fare to tighten the bound. Nevertheless, due to the gap between the fare for the quickest path and the cheapest fare, the searching size can still be large. Therefore, treatments are needed to accelerate the searching process. This paper suggests an effective way to include nonadditive fares.

2.2.2. Common Lines Problem. A line has a sequence of N stations and can be separated into $(N - 1)$ sections. Each line section consists of two successive stations with an arc connecting these two nodes. According to Tong and Richardson's algorithm, arcs with the same starting stations, end stations, and running times can be aggregated into a link with a unique transit time. The arcs of a link are listed in the time sequence. During the path search, for each link only the first arc having a departure time later than the train's arrival time at node i is used. However, this found path may not be the optimal path. An example of this problem is shown in Figure 1.

In the problem illustrated in Figure 1, the passenger arrives at A at 8:55 and wants to go to C. There are two train lines:

Line 1: it starts from A at 9:00 and ends at B at 10:00.

Line 2: it starts from A at 9:10, stops at B at 10:10, and finally ends at C at 11:00.

Hence, there are two possible paths:

Path 1: first use Line 1, and then transfer to Line 2.

Path 2: directly use Line 2.

These two paths have the same arrival time, but Path 2 has no transfer. Thus, Path 2 may be more attractive to passengers. However, Path 2 is not found when using Tong and Richardson's algorithm, because only one link is built between A and B, and the first arc that meets the time requirement belongs to Line 1. As this kind of line structure is common in railway systems, Tong and Richardson's algorithm needs to be modified if it is to be useful for these systems.

2.2.3. Waiting Time at Home. Tong and Wong [6] proved that all of the coefficients in the weighted cost function should be at least 1. If one of the coefficients is smaller than 1, the optimal path may not be found. However, the real situation is more complex. Passengers want to begin their trip at T_o^{DE}:

$$T_o^{\mathrm{DE}} = T_o^{\mathrm{ED}}, \tag{6}$$

where T_o^{ED} is the actual clock time for the earliest departure time at the origin o, unit: mins.

However, the passengers usually need to wait for a while, because the trains run based on their timetables. If the waiting time at the origin station exceeds the tolerance of passengers, t^{T}, the passengers will choose to leave their homes or offices later. A smaller penalty P^{WH} $(1 > P^{\mathrm{WH}} \geq 0)$ should be given for their time waiting at home or the office, because people can use that time for other activities. To consider this condition, a modification of Tong and Richardson's algorithm is needed.

This paper extends Tong and Richardson's algorithm to solve these limitations, so that the algorithm can better analyze dynamic railway networks.

3. Three Extensions

3.1. First Extension: Nonadditive Fare. To add the nonadditive fare into Tong and Richardson's algorithm, the total weighted cost should be modified first, as follows:

$$C_{iq}^{j} = f\left(t_{iq}^{j}\right) + S_{iq}^{j} * P^{\mathrm{T}} + g\left(c_{iq}^{j}\right), \tag{7}$$

where c_{iq}^{j} is the fare for path j from node i to q, unit: yuan, and $g(c_{iq}^{j})$ is the weighted function of fare converted to time, unit: mins. Also,

$$g\left(c_{iq}^{j}\right) = \frac{c_{iq}^{j}}{V}, \tag{8}$$

where V is the value of time, unit: yuan/min.

In addition, P^{WK}, P^{W}, P^{IV}, and P^{T} should not be less than 1 [6].

Second, a new step is needed before setting the upper boundary (i.e., Step 3).

A time-dependent cheapest path algorithm is developed from Dijkstra's algorithm [5], which uses fares between nodes as the path cost (based on the fare table) to search for the fare-dependent cheapest path from the origin o to the destination d. Thus, the smallest fare c_{od}^{S} can be found.

Third, the search boundary is tightened, as follows:

$$B_2 = C_{od}^{\mathrm{Q}} - g\left(c_{od}^{\mathrm{S}}\right) + T_o^{\mathrm{DE}}. \tag{9}$$

Proposition 1. *The optimal path's arrival time, T_d^{A-O}, is less than B_2.*

Proof. As P^{WK}, P^{W}, and P^{IV} should be at least 1, we have

$$t_{iq}^{j} \le f\left(t_{iq}^{j}\right). \tag{10}$$

Hence, for the optimal path O from the origin o to the destination d, we have

$$t_{od}^{O} \le f\left(t_{od}^{O}\right). \tag{11}$$

Adding the transfer penalty for the optimal path $S_{od}^{O} * P^{\mathrm{T}}$, the weighted fare function for the optimal path $g(c_{od}^{O})$, and T_{o}^{DE} to both the sides of the inequality, we have

$$\begin{aligned} & t_{od}^{O} + S_{od}^{O} * P^{\mathrm{T}} + T_{o}^{\mathrm{DE}} + g\left(c_{od}^{O}\right) \\ & \le f\left(t_{od}^{O}\right) + S_{od}^{O} * P^{\mathrm{T}} + g\left(c_{od}^{O}\right) + T_{o}^{\mathrm{DE}} \\ & \overset{\text{Equation (7)}}{\Longleftrightarrow} t_{od}^{O} + S_{od}^{O} * P^{\mathrm{T}} + T_{o}^{\mathrm{DE}} + g\left(c_{od}^{O}\right) \\ & \le C_{od}^{O} + T_{o}^{\mathrm{DE}} \\ & \overset{T_{d}^{\mathrm{A-O}} = t_{od}^{O} + T_{o}^{\mathrm{ED}}}{\Longleftrightarrow} T_{d}^{\mathrm{A-O}} + S_{od}^{O} * P^{\mathrm{T}} + g\left(c_{od}^{O}\right) \\ & \le C_{od}^{O} + T_{o}^{\mathrm{DE}} \end{aligned} \tag{12}$$

assuming that

$$T_{d}^{\mathrm{A-O}} > B_{2}. \tag{13}$$

Therefore, the above-given inequality can be changed to

$$\begin{aligned} & B_{2} + S_{od}^{O} * P^{\mathrm{T}} + g\left(c_{od}^{O}\right) < C_{od}^{O} + T_{o}^{\mathrm{DE}} \\ & \overset{\text{Equation (9)}}{\Longleftrightarrow} C_{od}^{\mathrm{Q}} - g\left(c_{od}^{\mathrm{S}}\right) + T_{o}^{\mathrm{DE}} + S_{od}^{O} * P^{\mathrm{T}} \\ & \qquad + g\left(c_{od}^{O}\right) < C_{od}^{O} + T_{o}^{\mathrm{DE}} \\ & \Longleftrightarrow C_{od}^{\mathrm{Q}} + S_{od}^{O} * P^{\mathrm{T}} + \left[g\left(c_{od}^{O}\right) - g\left(c_{od}^{\mathrm{S}}\right)\right] < C_{od}^{O}. \end{aligned} \tag{14}$$

For the definitions of the transfer penalty and the time-dependent cheapest path algorithm, we have

$$\begin{aligned} S_{od}^{O} * P^{\mathrm{T}} \ge 0, \\ g\left(c_{od}^{O}\right) - g\left(c_{od}^{\mathrm{S}}\right) \ge 0. \end{aligned} \tag{15}$$

Hence, the above inequality can be changed to

$$C_{od}^{\mathrm{Q}} < C_{od}^{O}. \tag{16}$$

This inequality shows that if the assumption is correct, the quickest path's total weighted cost is smaller than that of the optimal path. This result violates the definition of the optimal path, which should have the smallest total weighted cost. Hence, the optimal path's arrival time $T_{d}^{\mathrm{A-O}}$ is earlier than B_{2}.

Fourth, the optimal path search should be modified as well. There are two possible ways to add nonadditive fares.

One way is to add the fare to the weighted cost after the path search. When finding all of the possible paths, W_{od}^{j} is calculated. If $W_{od}^{j} > C_{od}^{\mathrm{Q}}$, the path search ends. After the path search, trace back the path and calculate $g(c_{od}^{j})$. Add $g(c_{od}^{j})$ to W_{od}^{j} to get C_{od}^{j}, and then compare all C_{od}^{j} to find the smallest value. The path that has the smallest value is the optimal path. The searching time for this process mainly depends on the difference between the fare of the quickest path and the cheapest fare. If this difference tends to be zero, then the boundary for searching is tight. Otherwise, the searching time may be very long.

The other way to add in the fare factor is to calculate the temporary cost ($C_{q}^{\mathrm{T}-j}$) along the path j search. When boarding a new line at node i, the cheapest ticket for this line from node i, c_{i*}^{s} is obtained from the fare table. Using this new line to arrive at node q, the temporary cost is

$$C_{q}^{\mathrm{T}-j} = \begin{cases} f\left(t_{oi}^{j}\right) + S_{oi}^{j} * P^{\mathrm{T}} + g\left(c_{oq}^{j}\right) + g\left(c_{q*}^{s}\right), & \text{staying onboard,} \\ f\left(t_{oq}^{j}\right) + S_{oq}^{j} * P^{\mathrm{T}} + g\left(c_{oq}^{j}\right), & \text{alighting.} \end{cases} \tag{17}$$

During the path search, $C_{q}^{\mathrm{T}-j}$ is compared with the total weighted cost of the present best path, C_{od}^{P}. This is the condition for the fare:

$$C_{od}^{P} \ge C_{q}^{\mathrm{T}-j}. \tag{18}$$

If the condition for the fare is not satisfied, the path search ends. Once path j reaches the destination, C_{od}^{j} has been calculated. If $C_{od}^{j} < C_{od}^{P}$, then path j is presently the best path, and the value of the upper boundary is replaced by $C_{od}^{j} - g(c_{od}^{\mathrm{S}}) + T_{o}^{\mathrm{DE}}$. □

Proposition 2. *If the condition of the fare is not satisfied, then path j from the origin o to node q cannot be the optimal path.*

Proof. We know that

$$C_{q}^{\mathrm{T}-j} \le C_{od}^{j}. \tag{19}$$

That is, if the condition of fare is not satisfied, that is,

$$C_{od}^{P} < C_{q}^{\mathrm{T}-j}, \tag{20}$$

then

$$C_{od}^{P} < C_{od}^{j}. \tag{21}$$

Thus, path j cannot be the optimal path. □

In this paper, the second approach is used to find the optimal path, as the second path provides a tighter bound for searching.

3.2. Second Extension: Common Lines Problem. The common lines problem can be solved by redefining the common lines. Here, the common lines are defined as those train lines having the same platform sequence, transit time, stopping time at each intermediate platform, and fare. A line has several arcs

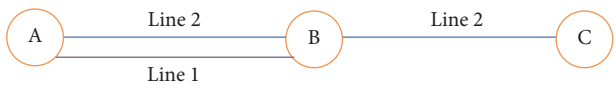

Figure 2: The rebuilt example network, using the new concept of common lines.

connecting each pair of two successive stations. Arcs with the same starting station, end station and running pattern for the whole line and fare, can be aggregated into a link with a unique transit time and fare. The arcs of the link are listed in the time sequence. To illustrate the difference between this new concept of common lines and Tong and Richardson's concept, the example network in the introduction is rebuilt, as shown in Figure 2.

Two links between A and B are built and searched. Paths 1 and 2 are identified and compared so the optimal path can be found. This example illustrates that, after using this new concept to build links, the optimal path can be found by selecting only the first train arc, which has a departure time later than the arrival time. The proof is presented in Proposition A.1 in Appendix.

This approach increases the searching size by adding more links, due to the stricter definition of common lines. Thus, a time window for searching train arcs is created to decrease the searching size. The lower boundary of the time window at node i is the arrival time at node i via the present path j, that is, T_i^{A-j}. Passengers using the present path j cannot take the train which departs at node i before T_i^{A-j}. The upper boundary of the time window at node i is the latest departure time to the destination d at node i, T_i^{LD}, calculated in the reversed path search. A similar concept has been suggested by Khani et al. [7], who used the quickest time from node i to the destination d (i.e., the sum of the transit time and walking time) to restrict the search. In fact, this restriction can be made even tighter by using the latest departure time and considering the waiting time, transit time, and walking time. A proof is given in Proposition A.2 in Appendix to show that using T_i^{LD} for the optimal path search is sufficient.

3.3. Third Extension: Departure Time Choice. As has been proven by Tong and Wong [6], the weighted coefficients for time must be at least 1, to ensure that the optimal path's ending time is lower than the upper boundary. However, when passengers stay at home or in their offices rather than at the station, they value that time as a relative benefit. Thus, it is reasonable to allow the waiting penalty at home or office (P^{WH}) to be smaller than 1.

First, a new concept, departure interval, is introduced to solve this problem. Usually, people tend to depart to the origin station at intervals such as 8:00, 8:15, or 8:30. The time intervals between each departure are defined as the departure interval, t^I. In our algorithm, we assume that $t^I \le t^T$. The algorithm divides the time window $[T_o^{ED}, T_d^{LA}]$ into different departure time sections, and each section's duration equals t^I, where T_d^{LA} is the latest arrival time at the destination d. For each departure time section at the origin station, only the trains departing at $[T_o^{DE}, T_o^{DE} + t^T]$ are considered. The maximum waiting time at the origin station is t^T, and the actual waiting time for each path is within the tolerance of the passengers. The waiting time at home is not considered during the path search of each time subsection, but this waiting time is considered after the path search so that the optimal path for each departure time section can be found. After searching all of the departure time sections, the optimal path for the time window $[T_o^{ED}, T_d^{LA}]$ can be found.

Another concept, that of the zone, can be introduced into the algorithm to simulate the condition in which passengers are traveling to or from the station. Zone links are built to connect zones and stations with a timetable for departures. Each zone link l has a unique journey time T_{zsl}^J between zone z and station s, and different journey times are used to simulate the congestion effects during the peak and off-peak periods. The penalty for a journey between zones and stations is P^{ZS} ($P^{ZS} \ge 1$).

Thus, a new process to find the optimal path is suggested and shown below.

Step 0. The line sections outside the time window $[T_h^{ED}, T_e^{LA}]$ do not need to be included in the searching network, and therefore a subnetwork can be created to increase the searching speed, where

T_h^{ED} is the earliest departure time at home zone h, unit: mins, and

T_e^{LA} is the latest arrival time at the end zone e, unit: mins.

Step 1. The passenger departs from home zone h at T_h^{DE}, and $T_h^{DE} = T_h^{ED}$.

Step 2. Run the quickest path search. In the origin station, only the trains departing at $[T_h^{DE} + T_{hoy}^J, T_h^{DE} + T_{hoy}^J + t^T]$ are considered, where T_{hoy}^J is the journey time of home zone link y between the home zone h and the origin station o.

Step 3. If no path is found, go to Step 11.

Step 4. The algorithm stops if $T_{en}^{A-Q} > T_e^{LA}$, where T_{en}^{A-Q} is the actual clock time for the arrival time at the end zone e, using the quickest path Q in the nth departure, unit: mins.

Go to Step 11 if $T_{en}^{A-Q} - T_h^{DE} > C'^P_{he}$, where C'^P_{he} is the present best path's weighted path cost, unit: mins.

The new weighted path cost function for path j is

$$C'^j_{he} = f\left(t_{od}^j\right) + S_{od}^j * P^T + g\left(c_{od}^j\right) + \left(T_{hoy}^J + T_{edw}^J\right) * P^{ZS}, \quad (22)$$

where T_{edw}^J is the journey time for end zone link w, between the end zone e and the destination station d, unit: mins.

Step 5. Run the cheapest path search. In the origin station, only those trains departing at $[T_h^{DE} + T_{hoy}^J, T_h^{DE} + T_{hoy}^J + t^T]$ are considered. Thus, the smallest fare, c_{od}^S, can be found.

Step 6. Calculate the new total weighted cost C'^{Q}_{he} and the upper boundary B_{he}, with the departure time for the subsection.

If the present best path's weighted path cost C'^{P}_{he} (which is initially infinite) is larger than C'^{Q}_{he}, then

$$B_{he} = C'^{Q}_{he} - g\left(c^{S}_{od}\right) + T^{DE}_{h}. \tag{23}$$

If not,

$$B_{he} = C'^{P}_{he} - g\left(c^{S}_{od}\right) + T^{DE}_{h}. \tag{24}$$

Step 7. Run the reversed quickest path search, using the boundary B_{he}. Compare T^{EA}_{i} and T^{LD}_{i} to check the accessibility of node i, A_i, by using (5). Exclude inaccessible nodes, so that these nodes will not be included in the optimal path search.

Step 8. Find the optimal path among the accessible nodes in the time window $[T^{DE}_{h}, B_{he}]$, and apply the following set of rules:

(1) In the origin station, only trains departing at $[T^{DE}_{h} + T^{J}_{hoy}, T^{DE}_{h} + T^{J}_{hoy} + t^{T}]$ are considered.

(2) Only use the first arc of link l, in which the departure time is later than the arrival time T^{A-j}_{i} at the starting node i of link l.

(3) If the arc of link l departs at node i later than T^{LD}_{i}, then the search for link l stops, as the arcs of link l are listed in the time sequence.

(4) In reality, passengers can only afford a certain number of transfers during their journeys. In considering this constraint, a transfer limit (L_{tran}) is introduced into the algorithm to stop the branch search if the number of transfers exceeds L_{tran}.

(5) If $T^{A-j}_{i} > T^{LD}_{i}$, the search branch stops.

(6) If the condition of fare limitation (i.e., (18)) is not satisfied, the search branch stops.

Step 9. Once the optimal path search ends, the waiting time at home should be added:

$$C^{j-H}_{he} = C'^{j}_{od} + (n-1) * t^{I} * P^{WH}, \tag{25}$$

where C^{j-H}_{he} is the total weighted cost of the newly found optimal path j from home zone h to end zone e, considering the waiting time at home or the office and the nonadditive fare, unit: mins.

Step 10. If the newly found optimal path has a lower total weighted cost than C^{P-H}_{he}, then record this path as the present best path and replace C^{P-H}_{he} as the new value. Note that, at the start of this process, the total weighted cost C^{P-H}_{he} equals the infinite.

Step 11. If $n = n + 1$ (and n is a positive integer), update the departure time:

$$T^{DE}_{h} = T^{ED}_{h} + (n-1) * t^{I}. \tag{26}$$

If $T^{DE}_{h} < T^{LA}_{e}$, go to Step 2. If $T^{DE}_{h} \geq T^{LA}_{e}$, the algorithm stops.

When all of the possible time sections have been searched, the path with the lowest weighted travel cost can be found. This path is the optimal path for the time window $[T^{ED}_{h}, T^{LA}_{e}]$.

We have tested this new algorithm using the data provided by Tong and Richardson [1]. However, fare data is not provided, so we set the fare to zero. In addition, as there is no further information about waiting at home, the penalty of this is the same as that of waiting at the platform. The example network of Tong and Richardson is simple, so the abovementioned common line problem does not exist. Our result is consistent with theirs, demonstrating that the new extensions do not violate the ability of the initial algorithm. In the next chapter, three examples are presented to show the advantages of the new extensions.

4. Case Studies

4.1. Description of the Example Network. To test the proposed algorithm, 16 important high-speed railway stations in southern China are selected to form an example network (shown in Figure 3). Except for Guangzhounan Station, which has three platforms, each of the other stations has two platforms (as shown in Figure 4). Each platform is considered as a node in the abstract network, and each station serves a zone. In addition, some basic assumptions regarding the passengers are set as follows:

The penalty for in-vehicle time is $P^{IV} = 1.0$.

The penalty for walking time is $P^{WK} = 2.0$.

The penalty for waiting time at the station is $P^{W} = 1.8$.

The penalty for transfers is $P^{T} = 1.0$.

The value of time is $V = 0.625$ yuan/min (assuming that the average salary is 6000 yuan/month and the working time is 160 hours/month).

The departure interval, t^{I}, is 15 mins.

The maximum waiting time at the origin station, t^{T}, is 15 mins.

The journey time T^{J}_{zsl} between a zone z and a station s is 5 mins.

The penalty for a journey between zone s and station is $P^{ZS} = 1$.

The transfer limit for passengers L_{tran} is 1.

These parameters can be obtained by using the state preference method based on real data.

Based on the example network, three extensions for the limitations and their respective examples are presented below.

Figure 3: Sixteen selected stations in southern China.

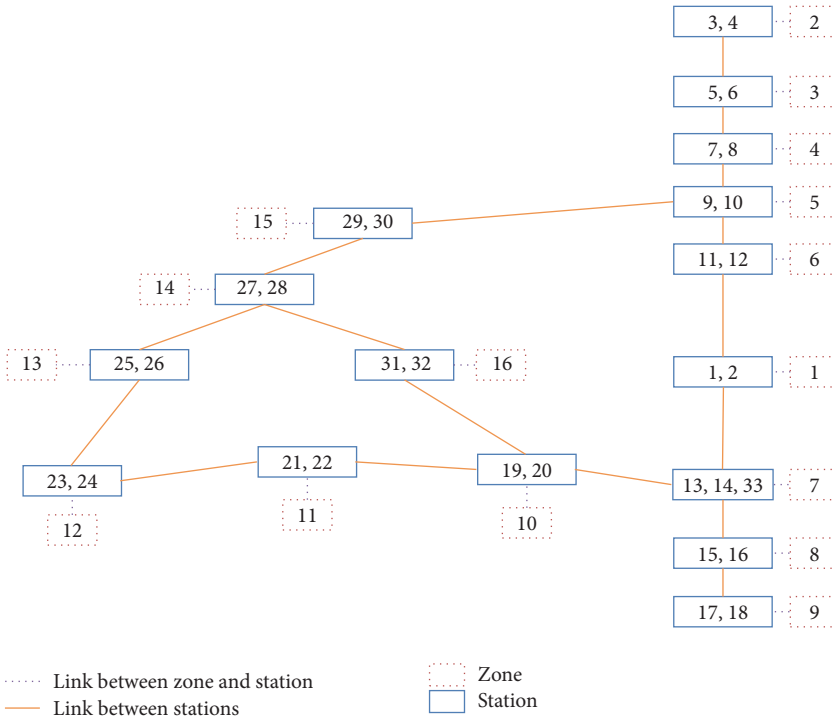

Figure 4: The abstracted network in southern China.

4.2. Example 1, for Extension 1. A passenger wants to travel from Guangzhou to Liuzhou, and the earliest departure time, T_o^{DE}, is 9:30. There are two possible paths, as shown in Tables 1 and 2, and their respective cost calculations are shown in Tables 3 and 4.

The quickest path is path j_1, which has a total weighted cost C_{od}^P of 626.2 mins and a cheapest fare of 185.5 yuan. Thus, the upper boundary is

$$9{:}30 + \lceil 626.2 \rceil - \left\lfloor \frac{185.5}{0.625} \right\rfloor = 15{:}01. \tag{27}$$

Table 1: The possible path j_1 for a trip from Guangzhou to Liuzhou.

Node	Arrival time	Departure time	Route used at arrival	Fare (yuan)
13	9:30	9:36		
19	10:21	10:23	Line 1	
31	11:18	11:20	Line 1	137.5
27	12:17	12:59	Line 1	
25	14:20	—	Line 2	48.0

Table 2: The possible path j_2 for a trip from Guangzhou to Liuzhou.

Node	Arrival time	Departure time	Route used at arrival	Fare (yuan)
13	9:30	9:33		
21	11:25	11:27	Line 3	185.0
23	14:55	14:55	Line 3	
24	14:56	14:56	Walk	
26	14:59	—	Line 4	1.0

Table 3: Cost calculation for the possible path j_1 for a trip from Guangzhou to Liuzhou.

	Actual value	Penalty	Weighted time
IVT-transit time	238	1.0	238.0
IVT-waiting time on train	4	1.0	4.0
Walking time	0	2.0	0.0
Waiting time at platform	48	1.8	86.4
Transfer	1	1.0	1.0
Fare	185.5	0.625	296.8
Sum			626.2

Table 4: Cost calculation for the possible path j_2 for a trip from Guangzhou to Liuzhou.

	Actual value	Penalty	Weighted time
IVT-transit time	323	1.0	323.0
IVT-waiting time on train	2	1.0	2.0
Walking time	1	2.0	2.0
Waiting time at platform	3	1.8	5.4
Transfer	1	1.0	1.0
Fare	186.0	0.625	297.6
Sum			631.0

The possible path j_2 is found, because its arrival time at destination is within the upper boundary, if the condition for the fare (i.e., (18)) does not need to be satisfied.

If the condition for the fare is used, the temporary cost is calculated along the path search (shown in Figure 5).

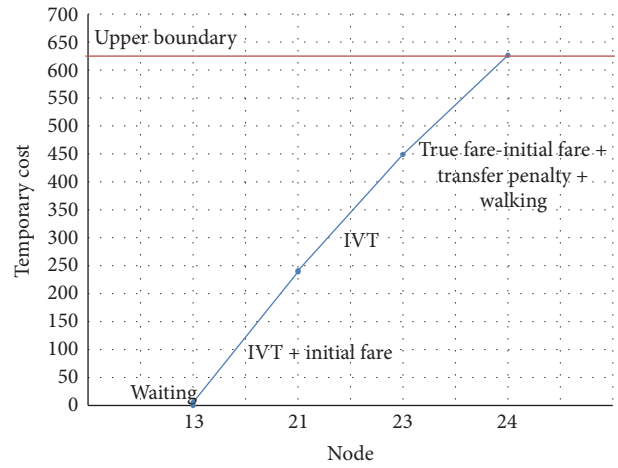

Figure 5: Temporary cost calculation along the possible path j_2.

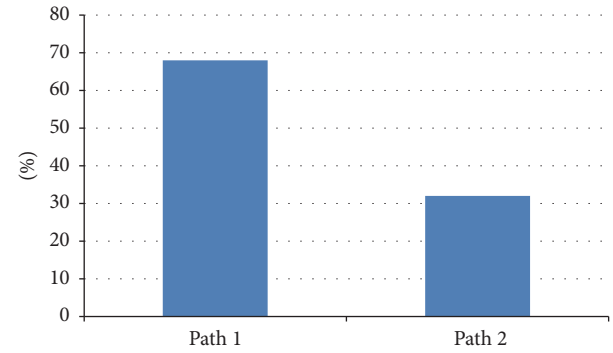

Figure 6: Test result in Example 1.

When the path search arrives at Node 24, using path j_2, the temporary cost is $626.4 > 626.2 = C_{od}^{P}$, as shown in Figure 5.

The path search for this section stops at Node 24. This example shows that the searching time is reduced if the condition for the fare is applied.

Both the characters of paths and the perceptions of passengers determine the optimal path. Considering this network structure, path j_1 has a shorter in-vehicle time and a cheaper fare, but the waiting time at the platform is much longer than the path j_2. The biggest advantage of path j_2 is therefore the short waiting time at the platform. Hence, we will use the Monte Carlo method to test the sensitivity of P^{W} under this circumstance.

We assume that P^{W} follows a normal distribution with a mean of 1.8 and a standard deviation of 0.2. A statistical tool based on the real data collected by the state preference method is used to obtain this information. We randomly generated 100 numbers that follow the distribution of P^{W} in Excel. While retaining the other parameters, we put these 100 numbers into the model one by one as the value of P^{W} and then ran the model to find the optimal path. The test result is shown in Figure 6. The path j_1 has a higher possibility of being selected due to its shorter in-vehicle time and cheaper fare. However, we also found that when P^{W} is larger than about

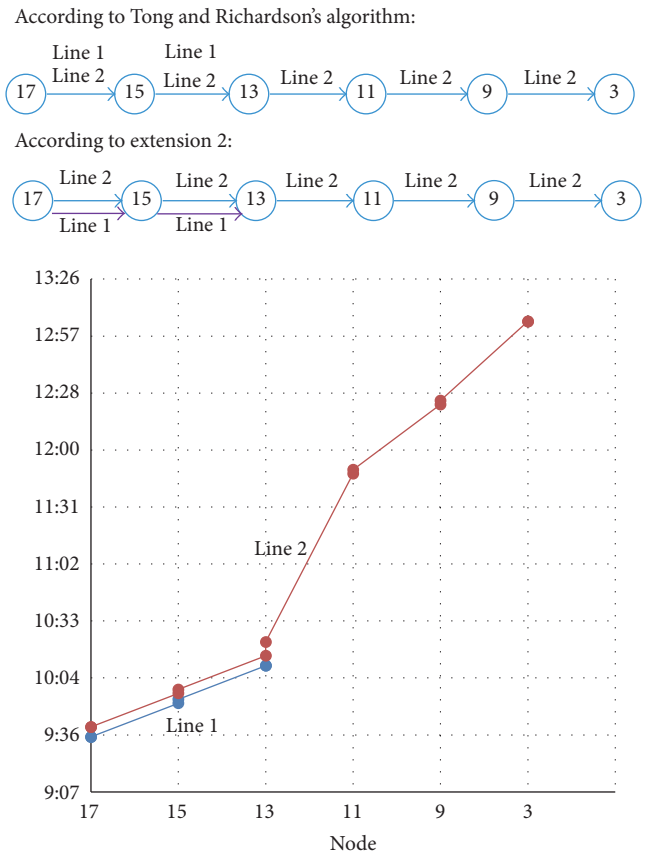

Figure 7: Two abstract networks in Example 2.

1.9, the optimal path is path j_2 rather than path j_1, so if a passenger really does not want to wait for a train, he or she is very likely to choose path j_2.

4.3. Example 2, for Extension 2. A passenger wants to travel from Shenzhen to Changsha, and the earliest departure time, T_o^{DE}, is 9:30. Two possible trains are given in Tables 5 and 6 and illustrated in Figure 7.

Based on Tong and Richardson's algorithm, a link from Node 17 to Node 15 should be created with two train numbers, that is, Line 1 and Line 2, with Line 1 used to travel from Shenzhenbei to Guangzhounan. Only the possible path j_1 can be found, as shown in Tables 7 and 8. However, based on the new concept of common lines, there should be two links, that is, one link associated with Line 1 and the other associated with Line 2. Another possible path, j_2, is found and is shown in Tables 9 and 10. The weighted cost for j_2 is smaller than that for j_1, so that the better path is j_2, which cannot be found by using Tong and Richardson's algorithm. In summary, this example shows that the first extension has a better solution than that provided by Tong and Richardson's algorithm.

4.4. Example 3, for Extension 3. A passenger wants to travel from Guangzhou to Liuzhou. The time window $[T_h^{ED}, T_e^{LA}] = [9:00, 18:00]$.

Table 5: The timetable of the high-speed railway Line 1 in Example 2.

Node	Station name	Arrival time	Departure time	Travel time between stations (mins.)
17	Shenzhenbei	—	09:35	
15	Humen	09:52	09:54	17
13	Guangzhounan	10:11	—	17

Table 6: The timetable of the high-speed railway Line 2 in Example 2.

Node	Station name	Arrival time	Departure time	Travel time between stations (mins.)
17	Shenzhenbei	—	09:40	
15	Humen	09:57	09:59	17
13	Guangzhounan	10:16	10:23	17
11	Chenzhouxi	11:48	11:50	85
9	Hengyangdong	12:23	12:25	33
3	Changshanan	13:05	—	40

Table 7: The possible path j_1 using Tong and Richardson's algorithm in Example 2.

Node	Arrival time	Departure time	Route used at arrival	Fare (yuan)
17	9:30	9:35		
15	9:52	9:54	Line 1	74.5
13	10:11	10:23	Line 1	
11	11:48	11:50	Line 2	
9	12:23	12:25	Line 2	314
3	13:05	—	Line 2	

Table 8: The possible path j_1 cost calculation in Example 2.

	Actual value	Penalty	Weighted time
IVT-transit time	192	1.0	192.0
IVT-waiting time on train	6	1.0	6.0
Walking time	0	2.0	0.0
Waiting time at platform	17	1.8	30.6
Transfer	1	1.0	1.0
Fare	388.5	0.625	621.6
Sum			851.2

In the first run, the actual departure time is 9:00. The quickest path's weighted path cost is 585 mins, and the cheapest fare is 185.5 yuan. Thus, the upper boundary, B_{he}, is

$$9{:}00 + 585 - \left\lfloor \frac{185.5}{0.625} \right\rfloor = 13{:}49. \quad (28)$$

Table 9: The possible path j_2 using the second extension in Example 2.

Node	Arrival time	Departure time	Route used at arrival	Fare (yuan)
17	9:30	9:40		
15	9:57	9:59	Line 2	
13	10:16	10:23	Line 2	388.5
11	11:48	11:50	Line 2	
9	12:23	12:25	Line 2	
3	13:05	—	Line 2	

Table 10: The possible path j_2 cost calculation in Example 2.

	Actual value	Penalty	Weighted time
IVT-transit time	192	1.0	192.0
IVT-waiting time on train	13	1.0	13.0
Walking time	0	2.0	0.0
Waiting time at platform	10	1.8	18.0
Transfer	0	1.0	0.0
Fare	388.5	0.625	621.6
Sum			844.6

Table 11: The possible path for the first run.

Node	Arrival time	Departure time	Route used at arrival
Zone 7	—	9:00	
14	9:05	9:12	Zone link
32	10:48	10:56	Line 1
28	12:01	12:01	Line 1
27	12:02	12:09	Walk
25	13:30	13:30	Line 2
Zone 13	13:35	—	Zone link

After the reversed quickest path search, a network including only accessible nodes can be built (as shown in Figure 8). One possible path is found in the optimal path search (shown in Tables 11 and 12). The present best weighted path cost is updated to 585 mins. As there is no waiting time at home, the present best total weighted cost is also 585 mins.

In the second run, the actual departure time is 9:15. No path is found.

In the third run, the actual departure time is 9:30. The quickest path's weighted path cost is 564.2 mins, and the cheapest fare is 185.5 yuan. Clearly, 564.2 mins is less than the previous best weighted path cost (of 585 mins). Thus, the upper boundary, B_{he}, is

$$9{:}30 + \lceil 564.2 \rceil - \left\lfloor \frac{185.5}{0.625} \right\rfloor = 13{:}59. \tag{29}$$

After the reversed quickest path search, a network having only accessible nodes can be built (as shown in Figure 9). One

Table 12: The possible path cost calculation for the first run.

	Actual value	Penalty	Weighted time
IVT-transit time	242	1.0	242.0
IVT-waiting time on train	8	1.0	8.0
Walking time	1	2.0	2.0
Waiting time at platform	14	1.8	25.2
Waiting time at home	0	0.5	0.0
Travel between zones and stations	10	1.0	10.0
Transfer	1	1.0	1.0
Fare	185.5	0.625	296.8
Sum			585.0

Table 13: The possible path for the third run.

Node	Arrival time	Departure time	Route used at arrival
Zone 7	—	9:30	
13	9:35	9:41	Zone link
19	10:26	10:28	Line 3
31	11:23	11:25	Line 3
27	12:22	12:24	Line 3
25	13:45	13:45	Line 4
Zone 13	13:50	—	Zone link

possible path is found in the optimal path search (as shown in Tables 13 and 14). The new path cost (579.2 mins) is more than the previous best total weighted cost, so the present best weighted path cost is updated to 579.2 − 15 = 564.2 mins.

In the fourth run, the actual departure time is 9:45. The quickest path's weighted path cost is 752 mins and the cheapest fare is 185.5 yuan. 752 mins is more than the previous best weighted path cost (564.2 mins). Thus, the upper boundary, B_{he}, is

$$9{:}45 + \lceil 564.2 \rceil - \left\lfloor \frac{185.5}{0.625} \right\rfloor = 14{:}14. \tag{30}$$

No path can be found. Using 752 mins, the upper boundary, B_{he}, is as follows:

$$9{:}45 + 752 - \left\lfloor \frac{185.5}{0.625} \right\rfloor = 17{:}21. \tag{31}$$

After the reversed quickest path search, a network having only accessible nodes can be built (as shown in Figure 10). One possible path is found in the optimal path search (as shown in Tables 15 and 16). The new path cost (774.5 mins) is more than the previous best weighted path cost, so this path cannot be the optimal path. This result indicates that tightening the upper boundary saves computation time.

The optimal path for this example is the possible path in the third run. This optimal path cannot be found by using Tong and Richardson's algorithm, because its arrival time (13:50) is later than the upper boundary of the first run (13:49).

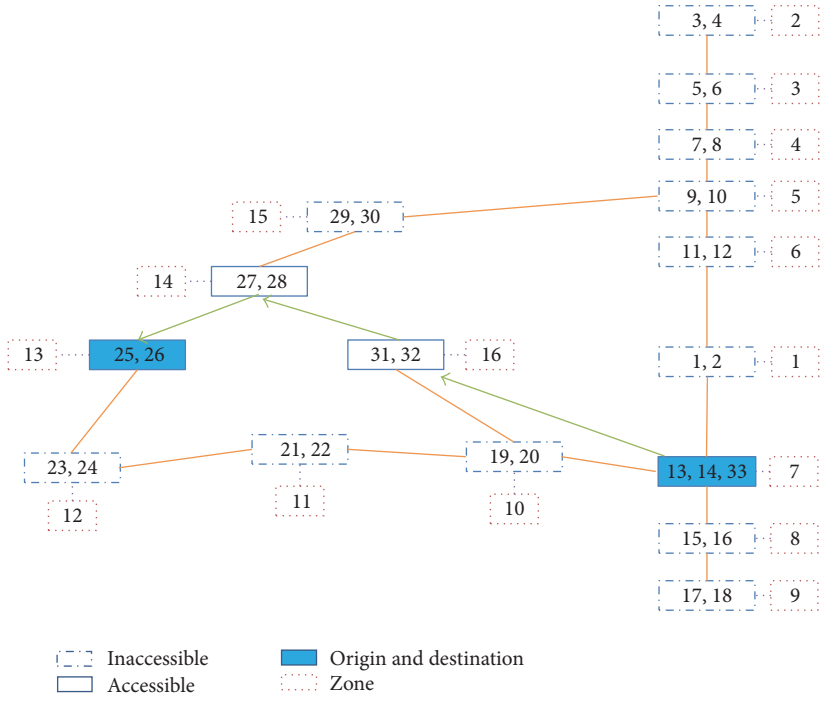

Figure 8: Network accessibility and the found optimal path for the first run.

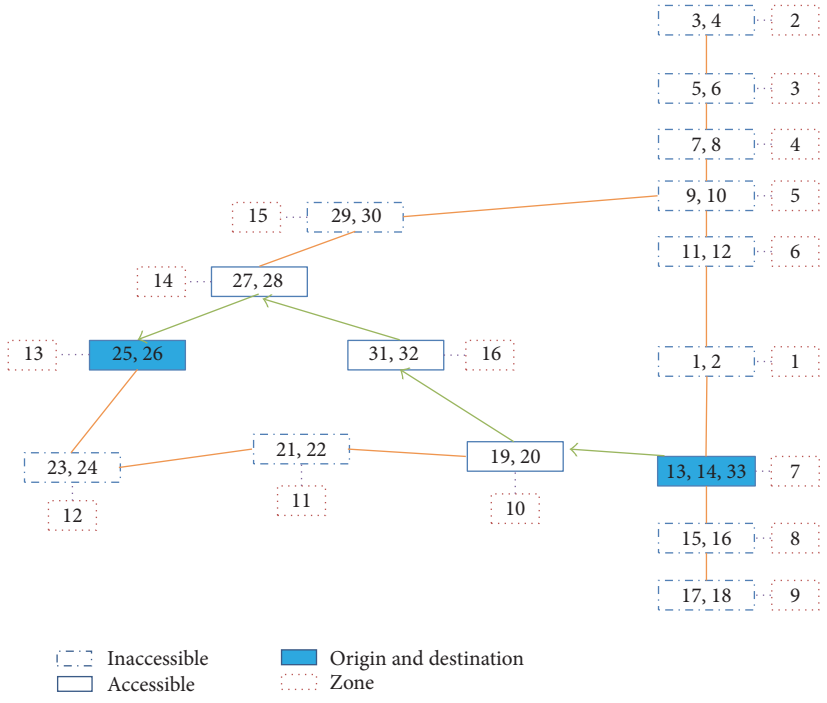

Figure 9: Network accessibility and the found optimal path for the third run.

5. Conclusions

This paper suggests three extensions for Tong and Richardson's algorithm. In the first, instead of only considering the time effect, the fare effect on choosing a path is included, so that the found path is more reasonable. In addition, a method for decreasing computation time is added. This method is demonstrated by using the cheapest fare and the temporary cost. In the second extension, the common lines problem in a schedule-based transit network is solved by redefining the common lines. The common lines should be those train lines that have the same platform sequences, transit times, and

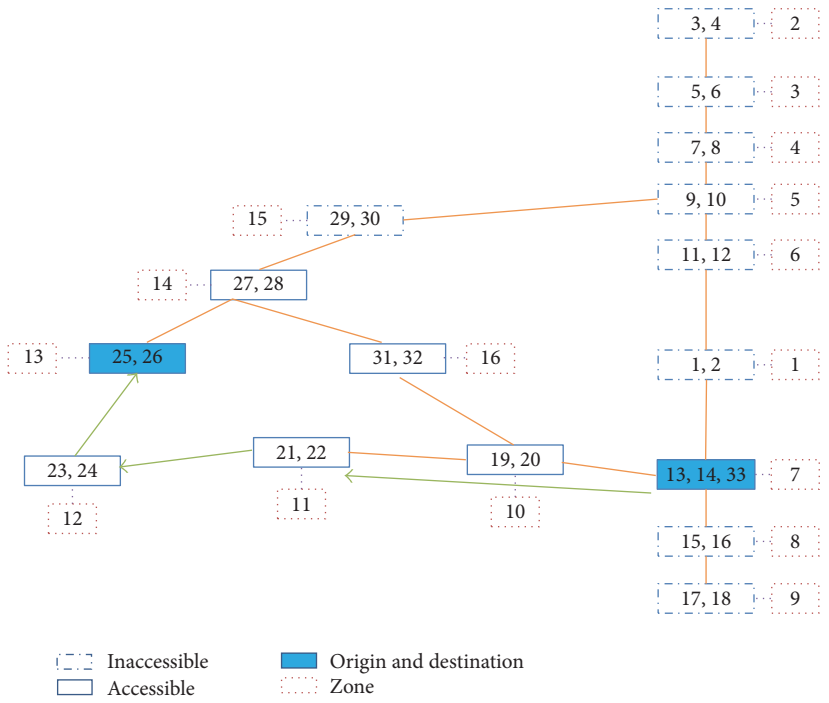

Figure 10: Network accessibility and the found optimal path using 752 mins in the fourth run.

Table 14: The possible path cost calculation for the third run.

	Actual value	Penalty	Weighted time
IVT-transit time	238	1.0	238.0
IVT-waiting time on train	4	1.0	4.0
Walking time	0	2.0	0.0
Waiting time at platform	8	1.8	14.4
Waiting time at home	30	0.5	15.0
Travel between zones and stations	10	1.0	10.0
Transfer	1	1.0	1.0
Fare	185.5	0.625	296.8
Sum			579.2

Table 15: The possible path for using 752 mins in the fourth run.

Node	Arrival time	Departure time	Route used at arrival
Zone 7	—	9:45	
13	9:50	9:53	Zone link
21	11:38	11:42	Line 5
23	13:49	14:19	Line 5
25	15:35	15:35	Line 6
Zone 13	15:40	—	Zone link

Table 16: The possible path cost calculation for using 752 mins in the fourth run.

	Actual value	Penalty	Weighted time
Transit time	308	1.0	308.0
Waiting time on train	4	1.0	4.0
Walking time	0	2.0	0.0
Waiting time at platform	33	1.8	59.4
Waiting time at home	45	0.5	22.5
Travel between zones and stations	10	1.0	10.0
Transfer	1	1.0	1.0
Fare	231	0.625	369.6
Sum			774.5

stopping times at each intermediate platform and fares. Line selecting rules are also added to reduce the computation complexity. In the third extension, departure time choice is introduced into the algorithm to avoid overly long waiting times at the origin station. This extension can also help to include the congestion effect in travel between the origin and destination stations. Hence, the new algorithm can be used to find the optimal path in a more complicated transit system, such as a real railway network. The examples shown demonstrate that the extended algorithm can work better than Tong and Richardson's algorithm. However, more extensions can be done, such as including the stochastic effects on the arrival times of passengers or trains, considering the capacity of trains or the fare change. Future

work will focus on these additional aspects. The uncertainty related to fares should be particularly considered, as the fare is an important factor in market competition, and rail companies are interested in how passengers change their behavior when the fare changes. Studies [19, 20] in this area have indicated some valuable new directions, and we will develop this algorithm further in the future.

Appendix

Propositions

Proposition A.1. *Based on the new concept of common lines, the optimal path via link l uses the first arc of link l having a departure time that is later than the arrival time.*

Proof. Assuming that a path j from the origin to node A must use link l_{AB} to arrive at node B and that l_{AB} has two train arcs that have departure times later than their arrival times at A, T_A^{A-j}, then one has the following:

Arc 1 belongs to Line 1, and its departing time at A is t_1.

Arc 2 belongs to Line 2, and its departing time at A is t_2.

If $T_A^{A-j} < t_1 < t_2$, then Arc 1 is the first arc having a departure time that is later than the arrival time.

There are two further conditions to consider:

Con 1: if passengers do not use Line 1 to arrive at A

Con 2: if passengers use Line 1 to arrive at A

There are two possible paths to take:

Path j_1: using Arc 1 to continue the journey

Path j_2: using Arc 2 to continue the journey

If there is no transfer after A, then the fare, transit time, and stopping time at each intermediate platform from A to the destination are the same for Lines 1 and 2, according to the new concept of common lines. In addition, path j_1 and path j_2 use the same arriving path to A. Thus, the total weighted cost difference between these paths is

$$\Delta C = \begin{cases} \left[\left(t_2 - T_A^{A-j}\right) - \left(t_1 - T_A^{A-j}\right)\right] * P^W, & \text{Con 1,} \\ \left[\left(t_2 - T_A^{A-j}\right) * P^W - \left(t_1 - T_A^{A-j}\right) * P^{IV}\right] + P^T, & \text{Con 2.} \end{cases} \quad \text{(A.1)}$$

The result of this equation is valid because

$$t_2 - T_A^{A-j} > t_1 - T_A^{A-j}, \quad \text{(A.2)}$$

$$P^W \geq P^{IV} \geq 1, \quad \text{(A.3)}$$

$$P^T \geq 1. \quad \text{(A.4)}$$

It can be seen that $\Delta C > 0$.

If a transfer happens after A, then an assumption can be made. Suppose that path j_1 and path j_2 have their first transfers at node C and they have another set of common lines after A. In that case,

Path j_1: the departing time at A is t_1, and the arriving time at C is t_3;

Path j_2: the departing time at A is t_2, and the arriving time at C is t_4.

If $t_1 < t_2$, then $t_3 < t_4$, because the transit time and stopping time at each intermediate platform from A to C are the same for path j_1 and path j_2, according to the new concept of common lines. At C, Arc 3 is the first arc which departs at C later than t_3, and the departure time of Arc 3, t_5, is also assumed to be later than t_4. Thus, the total weighted cost difference between path j_1 and path j_2 is as follows:

$$\Delta C = \begin{cases} \left[\left(t_2 - T_A^{A-j} + t_5 - t_4\right) - \left(t_1 - T_A^{A-j} + t_5 - t_3\right)\right] * P^W, & \text{Con 1,} \\ \left[\left(t_2 - T_A^{A-j} + t_5 - t_4\right) - \left(t_5 - t_3\right)\right] * P^W - \left(t_1 - T_A^{A-j}\right) * P^{IV} + P^T, & \text{Con 2.} \end{cases} \quad \text{(A.5)}$$

Therefore,

$$\Delta C = \begin{cases} \left[\left(t_2 - t_4\right) - \left(t_1 - t_3\right)\right] * P^W, & \text{Con 1,} \\ \left(t_2 - T_A^{A-j} - t_4 + t_3\right) * P^W - \left(t_1 - T_A^{A-j}\right) * P^{IV} + P^T, & \text{Con 2.} \end{cases} \quad \text{(A.6)}$$

According to the new concept of common lines,

$$t_2 - t_4 = t_1 - t_3. \quad \text{(A.7)}$$

Then, $\Delta C = 0$, for Con 1.

As for Con 2, the results from (A.3) and (A.4) indicate that

$$\Delta C = \left(t_2 - T_A^{A-j} - t_4 + t_3\right) * P^W - \left(t_1 - T_A^{A-j}\right) * P^{IV} + P^T$$

$$\geq \left(t_2 - T_{\mathrm{A}}^{\mathrm{A}-j} - t_4 + t_3\right) * P^{\mathrm{W}} - \left(t_1 - T_{\mathrm{A}}^{\mathrm{A}-j}\right) * P^{\mathrm{W}} + P^{\mathrm{T}} = \left(t_2 - t_4 + t_3 - t_1\right) * P^{\mathrm{W}} + P^{\mathrm{T}} > 0. \tag{A.8}$$

It is possible that t_5 is smaller than t_4; that is, path j_2 needs to use the next arc, which departs at C later than t_4. In that case, this path is called Arc 4, with its departure time at C being t_6. Also, the total weighted cost difference between path j_1 and path j_2 is

$$\Delta C = \begin{cases} \left[\left(t_2 - T_{\mathrm{A}}^{\mathrm{A}-j} + t_6 - t_4\right) - \left(t_1 - T_{\mathrm{A}}^{\mathrm{A}-j} + t_5 - t_3\right)\right] * P^{\mathrm{W}}, & \text{Con 1,} \\ \left[\left(t_2 - T_{\mathrm{A}}^{\mathrm{A}-j} + t_6 - t_4\right) - \left(t_5 - t_3\right)\right] * P^{\mathrm{W}} - \left(t_1 - T_{\mathrm{A}}^{\mathrm{A}-j}\right) * P^{\mathrm{IV}} + P^{\mathrm{T}}, & \text{Con 2.} \end{cases} \tag{A.9}$$

Similarly,

$$\begin{aligned} &\Delta C = \left(t_6 - t_5\right) * P^{\mathrm{W}}, \quad \text{Con 1,} \\ &\Delta C \geq \left(t_6 - t_5\right) * P^{\mathrm{W}} + P^{\mathrm{T}}, \quad \text{Con 2.} \end{aligned} \tag{A.10}$$

It can be seen that $\Delta C > 0$.

Thus, path j_1 is either better than or the same as path j_2; that is, using path j_1 can ensure the best result. Therefore, based on the new concept of common lines, the optimal path via link l uses the first arc of link l having a departure time later than the arrival time. □

Proposition A.2. *The optimal path does not use a train that departs from node i later than* T_i^{LD}.

Proof. Assuming that train k departs from node i at T_{ki}^{T} and that $T_{ki}^{\mathrm{T}} > T_i^{\mathrm{LD}}$, then the train arrives at the destination d at $B_2 + \Delta t$. According to the reverse path algorithm, we know that $\Delta t > 0$. If not, then T_i^{LD} should be T_{ki}^{T}. According to Proposition 1, no optimal path's arrival time $T_d^{\mathrm{A}-O}$ can be later than B_2. Hence, the optimal path does not use a train which departs from node i later than T_i^{LD}.

Furthermore, if path j can be the optimal path, then according to Proposition A.2 we have

$$T_i^{\mathrm{A}-j} \leq T_i^{\mathrm{D}-j} = T_{ki}^{\mathrm{T}} \leq T_i^{\mathrm{LD}}, \tag{A.11}$$

where $T_i^{\mathrm{D}-j}$ is the departure time at node i via path j, unit: mins. □

Notations

A_i: The accessibility of node i
B_1: The upper boundary, $W_{od}^{\mathrm{Q}} + T_o^{\mathrm{DE}}$, unit: mins
B_2: The upper boundary, $C_{od}^{\mathrm{Q}} - g(c_{od}^{\mathrm{S}}) + T_o^{\mathrm{DE}}$, unit: mins
B_{he}: The upper boundary of the extended algorithm, unit: mins
c_{iq}^j: The fare for path j from node i to q, unit: yuan
c_{od}^{S}: The smallest fare from the origin o to the destination d, unit: yuan
c_{q*}^s: The smallest ticket fare for the present line from the last transfer node q, unit: yuan
C_{iq}^j: The weighted cost for path j from node i to q, unit: mins
C'^{j}_{he}: The new weighted cost for path j from home zone h to end zone e, unit: mins
$C_{he}^{j-\mathrm{H}}$: The total weighted cost of path j from home zone h to end zone e, considering the waiting time at home or office and the nonadditive fare, unit: mins
C_j^{T}: The temporary cost along the path j, unit: mins
d: The destination
e: The end zone
$f(t_{iq}^j)$: The weighted function for journey time using path j from node i to q, unit: mins
$g(c_{iq}^j)$: The weighted function of fare using path j from node i to q, converted to time, unit: mins
h: The home zone
i or q: The identification for nodes
j: The identification for paths
k: The identification for trains
l or y, w: The identification for links
L_{tran}: The maximum number of transfers for a journey
o: The origin
O: The optimal path departing from the origin o to the destination d at T_o^{DE}
P: The present best path departing from the origin o to the destination d at T_o^{DE}
P^{IV}: The penalty for in-vehicle time, $P^{\mathrm{IV}} \geq 1$
P^{T}: The penalty for transfer, $P^{\mathrm{T}} \geq 1$, unit: mins
P^{W}: The penalty for waiting time, $P^{\mathrm{W}} \geq 1$
P^{WH}: The penalty for waiting at home or office, $1 > P^{\mathrm{WH}} \geq 0$
P^{WK}: The penalty for walking time, $P^{\mathrm{WK}} \geq 1$
P^{ZS}: The penalty for a journey between zones and stations, $P^{\mathrm{ZS}} \geq 1$

Q: The quickest path departing from the origin o to the destination d at T_o^{DE}
s: The identification for stations
S_{iq}^{j}: The number of transfers for path j from node i to q
t^{I}: The departure interval, unit: mins
t_j^{IV}: The in-vehicle time for path j, unit: mins
t_{iq}^{j}: The journey time, including waiting time, walking time, and transit time for path j from node i to q, unit: mins
t^{T}: The tolerance of passengers for waiting at the origin station, unit: mins
t_j^{WK}: The walking time for path j, unit: mins
t_j^{W}: The waiting time at stations for path j, unit: mins
$T_i^{\mathrm{A}-j}$: The arrival time at node i via path j, unit: mins
$T_{en}^{\mathrm{A}-Q}$: The actual clock time for the arrival time at the end zone e, using the quickest path Q in the nth departure, unit: mins
T_{ki}^{T}: The actual clock time when train k departs at node i, unit: mins
T_i^{EA}: The earliest arrival time at node i if departing from the origin o at T_o^{DE}, unit: mins
T_i^{EAR}: The earliest arrival time at node i from the destination d in the reversed network, unit: mins
T_h^{ED}: The earliest departure time at home zone h, unit: mins
T_o^{ED}: The actual clock time for the earliest departure time at the origin o, unit: mins
T_h^{DE}: The actual clock time for the departure time at home zone h, unit: mins
T_o^{DE}: The actual clock time for the departure time at the origin o, unit: mins
$T_i^{\mathrm{D}-j}$: The departure time at node i via path j, unit: mins
T_d^{LA}: The latest arrival time at the destination d, unit: mins
T_e^{LA}: The latest arrival time at the end zone e, unit: mins
T_i^{LD}: The latest departure time to the destination d at node i, unit: mins
T_{zsl}^{J}: The journey time of zone link l between a zone z and a station s, unit: mins
V: The value of time, unit: min/yuan
W_{iq}^{j}: The weighted cost for path j from node i to q, used in Tong and Richardson's algorithm, unit: mins
z: The identification for zones.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The work described in this paper was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Project no. [T32-101/15-R]). The first author was also supported by two scholarships (a Postgraduate Scholarship from the University of Hong Kong and a University Postgraduate Fellowship from the Jessie and George Ho Charitable Foundation).

References

[1] C. O. Tong and A. J. Richardson, "A computer model for finding the time-dependent minimum path in a transit system with fixed schedules," *Journal of Advanced Transportation*, vol. 18, no. 2, pp. 145–161, 1984.

[2] C. Chriqui and P. Robillard, "Common bus lines," *Transportation Science*, vol. 9, no. 2, pp. 115–121, 1975.

[3] S. Nguyen and S. Pallottino, "Equilibrium traffic assignment for large scale transit networks," *European Journal of Operational Research*, vol. 37, no. 2, pp. 176–186, 1988.

[4] H. Spiess and M. Florian, "Optimal strategies: a new assignment model for transit networks," *Transportation Research Part B: Methodological*, vol. 23, no. 2, pp. 83–102, 1989.

[5] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[6] C. O. Tong and S. C. Wong, "Minimum path algorithms for a schedule-based transit network with a general fare structure," in *Schedule-Based Dynamic Transit Modeling: Theory and Applications*, N. H. M. Wilson and A. Nuzzolo, Eds., vol. 28 of *Operations Research/Computer Science Interfaces Series*, pp. 241–261, Springer, New York, NY, USA, 2004.

[7] A. Khani, M. Hickman, and H. Noh, "Trip-based path algorithms using the transit network hierarchy," *Networks and Spatial Economics*, vol. 15, no. 3, pp. 635–653, 2015.

[8] W. Xu, S. He, R. Song, and S. S. Chaudhry, "Finding the K shortest paths in a schedule-based transit network," *Computers and Operations Research*, vol. 39, no. 8, pp. 1812–1826, 2012.

[9] M. Friedrich, I. Hofsäß, and S. Wekeck, "Timetable-based transit assignment using branch and bound techniques," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1752, pp. 100–107, 2001.

[10] R. Huang and Z.-R. Peng, "Schedule-based path-finding algorithms for transit trip-planning systems," *Transportation Research Record*, no. 1783, pp. 142–148, 2002.

[11] M. Friedrich and S. Wekech, "A schedule-based transit assignment model addressing the passengers' choice among competing connections," *Computer Science Interfaces Series*, vol. 28, pp. 159–173, 2004.

[12] R. Huang, "A schedule-based pathfinding algorithm for transit networks using pattern first search," *GeoInformatica*, vol. 11, no. 2, pp. 269–285, 2007.

[13] S. Y. Zhu, Y. F. Yan, H. Wang, and S. B. Li, "An optimal transit path algorithm based on the terminal walking time judgment and multi-mode transit schedules," in *Proceedings of the International Conference on Intelligent Computation Technology and Automation (ICICTA '10)*, pp. 623–627, Changsha, China, May 2010.

[14] D. Canca, A. Zarzo, P. L. González-R, E. Barrena, and E. Algaba, "A methodology for schedule-based paths recommendation

in multimodal public transportation networks," *Journal of Advanced Transportation*, vol. 47, no. 3, pp. 319–335, 2013.

[15] M. Florian, "Finding shortest time-dependent paths in schedule-based transit networks: a label setting algorithm," in *Schedule-Based Dynamic Transit Modeling: Theory and Applications*, N. H. M. Wilson and A. Nuzzolo, Eds., vol. 28 of *Operations Research/Computer Science Interfaces Series*, pp. 43–52, Springer, New York, NY, USA, 2004.

[16] O. A. Nielsen and R. D. Frederiksen, "Large-scale schedule-based transit assignment-further optimization of the solution algorithms," in *Schedule-Based Modeling of Transportation Networks*, pp. 1–26, Springer, Berlin, Germany, 2009.

[17] O. A. Nielsen, O. Landex, and R. D. Frederiksen, "Passenger delay models for rail networks," in *Schedule-Based Modeling of Transportation Networks*, pp. 1–23, Springer, Berlin, Germany, 2009.

[18] Y.-L. Chen and H.-H. Yang, "Finding the first K shortest paths in a time-window network," *Computers and Operations Research*, vol. 31, no. 4, pp. 499–513, 2004.

[19] C.-W. Yang and C.-C. Chang, "Applying price and time differentiation to modeling cabin choice in high-speed rail," *Transportation Research Part E: Logistics and Transportation Review*, vol. 47, no. 1, pp. 73–84, 2011.

[20] L. Deng, Z. Zhang, K. Liu, W. Zhou, and J. Ma, "Fare optimality analysis of urban rail transit under various objective functions," *Discrete Dynamics in Nature and Society*, vol. 2014, Article ID 910736, 8 pages, 2014.