An Augmented Algorithm for Public Transit Transfers

Wei Sun
SDUST
Qingdao,China, 266590
83391860@qq.com

Lining Zhu
CASM
Beijing, China, 100830
1446181280@qq.com

Zhiyuan Hong
CASM
Beijing, China, 100830
290475826@qq.com

Abstract

In this work, we have proposed an optimized algorithm for public transit transfers based on the principle of least transfers. The first augmentation introduced by this algorithm concerns the configuration of proximal distances from the start and end points: the proximal distance from the starting point has been set to a user-defined maximum permissible walking distance, and the proximal distance from the end point was allowed to vary dynamically. Second, the one-to-one start-to-end point correspondence of conventional search models was improved by replacing it with a one-to-many correspondence. Third, two threshold values were introduced to constrain the search process, and optimal values were obtained for these thresholds via statistical analyses on large quantities of experimental data. Finally, an experiment was performed using the public transit data of Changzhi City in the Shanxi Province to validate the effectiveness and viability of the augmented algorithm. The results of this experiment demonstrate that our augmented algorithm effectively improves the practicality of routes calculated by least transfer algorithms, and reduces search times.

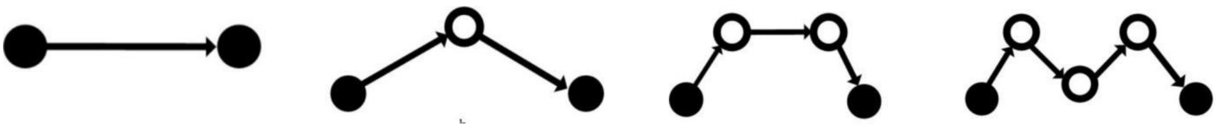
Urban public transport systems are very large, complex and open systems that integrate a city's transportation system with the socio-economic needs of urban societies. The development of public transportation is a developmental trend in many countries around the world, and research on urban public transportation systems is a topic of keen interest for many cities in China and elsewhere [1]. The continued development and improvement of urban public transportation services has made public transit increasingly convenient, flexible and accessible, but the increasing complexity of transit networks has also led to difficulties in the selection of routes and transfers.

Current mainstream algorithms for solving the public transit transfer problem include the least transfer algorithm, Dijkstra algorithm, Floyd algorithm, and A* algorithm. The least transfer algorithm in particular, is a good fit for easy acceptance by public transit users. Many researchers have worked to improve this algorithm, resulting in a considerable number of research achievements. The algorithms proposed by References [2 –7] are augmented least transfer algorithms that include proximal station judgments and walking factors in their analyses, and these algorithms generally produce routes and transfers that are a better fit for the actual selection scenarios of public transit users. Some scholars have also proposed effective measures for resolving the efficiency problem of least transfer algorithms. For example, Liao et al. [2] used the advantages of databases in rapid queries, indexing support and set operations to improve the efficiency of least transfer algorithms. Fu [3] combined the indexing and rapid query capabilities of databases and spatial database engines with high-efficiency processing mechanisms such as internal storage-based queries and set operations, which effectively enhanced the efficiency of least transfer algorithms.

Chen et al. [4] improved the efficiency and accuracy of public transit transfer algorithms by constructing integration relationships between the geometry and semantic connectivity of urban transportation networks. Liu [5] limited the area of search regions to improve search efficiency. Chang et al. [6] extracted key transfer points through the interconnectivity of public transit networks and the spatial/positional relationships between transit stops, thus improving search efficiency. Wang et al. [7] accounted for walking during intermediary transfers, and generated arrays for walking transfers via data preprocessing, which reduced search times. Nonetheless, these augmented least transfer algorithms still suffer from certain flaws. First, all currently existing algorithms define the proximal distance from start and end points (i.e., the range for the selection of proximal stations) as fixed values. For example, References [5] and [7] define this as 300 m and 500 m, respectively. Setting too small a value could lead to the exclusion of the optimal proximal station, whereas excessively large values will tend to result in walking distances that exceed the range acceptable to a user. Second, the improvements in efficiency delivered by these augmented algorithms remain insufficient for meeting general usage requirements.

To address the inadequacies highlighted above, we propose an optimized algorithm for public transit transfers. This algorithm abolishes the use of fixed values for start and end point proximal distances, and sets them as user-defined values instead. In addition, the proximal distance from end points was allowed to vary dynamically, which effectively avoids the exclusion of optimal proximal stations and reduces the occurrence of start/end point walking distances that exceed the tolerances of a user. A one-to-many correspondence was also used in lieu of the one-to-one start-to-end-point correspondence used by conventional search models, and two threshold values were included to regulate the search process. These changes have moderately improved the efficiency of the least transfer algorithm.

In accordance with the travel preferences of public transit users, the least transfer algorithm will first check whether the end point can be directly reached from the starting point during the selection of transit routes, as shown in Figure 1(a). If this is not possible, the algorithm will check whether an end point can be reached after one transfer, as shown in Figure 1(b). The set of stops that can be reached by the starting point will be searched for, and if there is a stop in this set that directly connects to the end point, it is then possible to reach the end point via one transfer. If a suitable travel plan cannot be found, the algorithm will check whether the end point can be reached via two transfers, as shown in Figure 1(c). The set of stops that can be reached from the starting point will be searched for, as well as the set of stops that can be reached from the end point. If any of the stops in the former set can directly connect to a stop in the latter set, it is then possible to travel from the starting point to the end point via two transfers. If a suitable plan still has not been found, the algorithm will search for the set of stops that can be reached from the starting point after one transfer, and the set of stops that can be reached from the end point. If any of the stops in the former set can directly reach a stop in the latter set, one may then travel from the starting point to the end point via three transfers, as shown in Figure 1(d). If this is still not possible, four transfers or more are then necessary for reaching the end point.



(a) (b) (c) (d)

Figure 1: The least transfer algorithm.

In our augmented algorithm, we use dynamic determination of start and end points to address the flaws of current least transfer algorithms in the configuration of start/end point proximal distances. This algorithm also addresses problems related to search efficiency and travel time when walking is taken into account, by replacing the one-to-one correspondence between start and end points with a one-to-many correspondence, thus preventing redundant searches and reducing search time. In addition, two threshold values will be added to regulate the search process: the number of start and end points to be prioritized in the search process (nearNum) and the maximum number of cases (maxCases). This ensures the efficiency of the search process, and limits its time consumption. The procedure of the augmented algorithm is as follows:

Step 1: Enter the name or coordinates of the stop.

Step 2: Obtain the sets of start and end points corresponding to the added proximal stops.

Step 3: Cyclically read the starting points and the corresponding sets of end points, and use the direct arrival model to search for transit plans.

Step 4: Decide whether the number of plans that have been found is smaller than maxCases; execute the next step if this is true; otherwise stop the search.

Step 5: Cyclically read the first nearNum stop in the set of starting points and first nearNum stop in the corresponding sets of end points, and use the 1-transfer model to search for plans. End this search when the number of plans found by the search process has reached maxCases.

Step 6: Determine whether the number of plans that have been found is 0. Execute the next step if this is true; otherwise end the search.

Step 7: Search for plans using the 1-transfer model from the start and end points found in Step 5 that have not participated in the 1-transfer search, and end the search when the number of plans has reached maxCases.

Step 8: Determine whether the number of plans is 0. Execute the next step if this is true, and end the search otherwise.

Step 9: Use the 2-transfer model to search for plans, using the same search procedure as the 1-transfer search.

Step 10: If the number of plans is 0, use the 3-transfer model to search for plans, using the same search procedure as the 1-transfer search. End the search once the number of plans found by this search has reached the user-defined number.

Step 11: End the search.

3.1. Dynamically determined starting points and end points

If the proximal distance was set to too small a value, an optimal proximal stop may be excluded by the algorithm. If the proximal distance is too large, many start and end point pairs that require walking distances exceeding the acceptable range of a user will be included in the possible transit plans. Furthermore, the participation of these start and end points in the search will also create wastage in terms of search time, and will necessitate a plan screening procedure that will lead to further time wastage. The dynamic determination of start and end points was therefore proposed to solve this issue.

The dynamic determination of start and end points simply means that the proximal distance from the end point is not a fixed value, but is instead a value determined by the proximal distance from the starting point. If De represents the proximal distance from the end point, Ds represents the proximal distance from the starting point, and MaxWalkDis represents the maximum walking distance that is acceptable by a user, the equation for calculating the proximal distance from the end point, De, is then

$$De=MaxWalkDis-Ds \qquad (1)$$

When searching for proximal stops from the starting point, MaxWalkDis is used as the search distance. The distance from the starting point to each proximal station, Dr, is then added to Ds one after another. Equation (1) is used to calculate the proximal distance from the end point, De, and proximal stations that are within De of the end point are also treated as end points. A set of end points is thus formed (as indicated by the dotted lines in Figure 2), and these end points correspond to proximal stations within a distance of Ds from the starting point. The proximal stops of the starting point are also treated as starting points, thus forming a set of starting points (as shown in Figure 2). Each stop in the set of starting points corresponds to a set of end points. In addition, the actual distances between the start and end points and their proximal stops (Dr) are also recorded. The elements in the starting and end point sets are then ordered according to the value of Dr, so that the stops closer to the original starting and end points are prioritized in the search process.

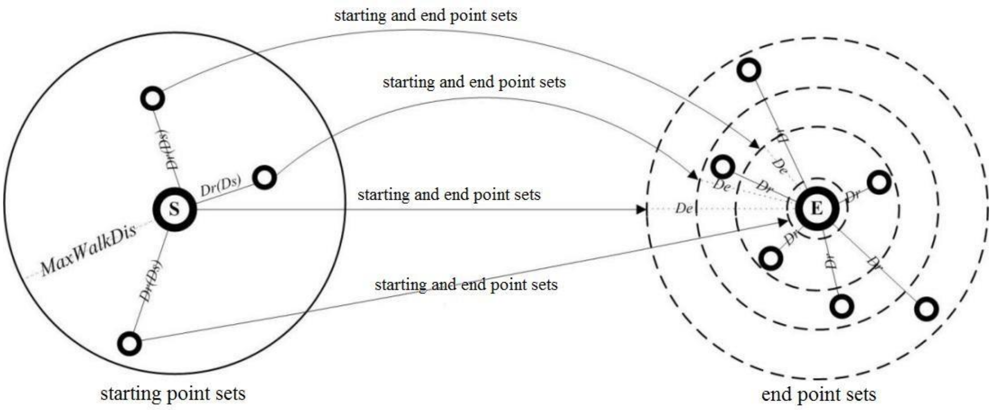
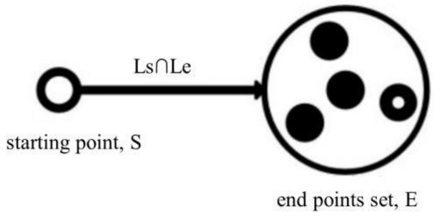


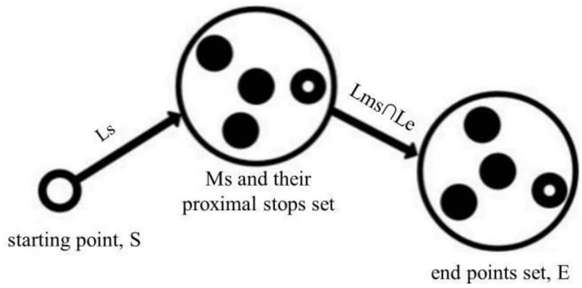
Figure 2: The starting point and end point sets.

3.2. The 1-to-N search model

Most existing least transfer algorithms use a one-to-one start-to-end-point correspondence in their search models. In our algorithm, a single starting point corresponds to a set of end points, i.e., N end points, where N is a positive integer. The reason we have proposed this one-to-N search model is because the one-to-one model leads to redundant searches for some of the data, and our experiments prove that the one-to-N model is superior to the one-to-one model in terms of search speeds.

Based on empirical evidence, two transfers or fewer will be sufficient for satisfying the requirements of most public transit users after walking distances have been accounted for. There are very few cases that require plans with three or more transfers, and one of the most significant flaws of the least transfer algorithm is that the search speed will become slower with increases in the number of transfers. Therefore, we have temporarily ignored plans with three or more transfers, and constructed four search models defined by the number of transfers: the direct arrival (0-transfer) model, 1-transfer model, 2-transfer model and 3-transfer model.

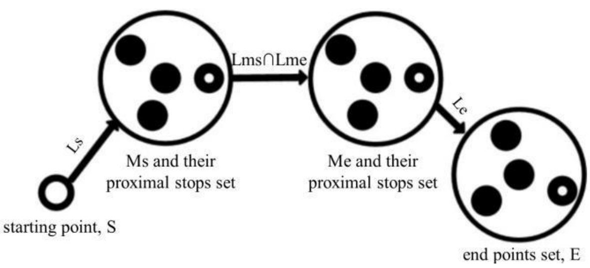


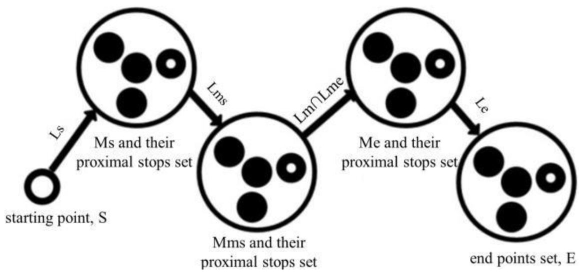


(a) Direct arrival (0 transfers)

(b) One transfer

(b)





(c) Two transfers

(d)Three transfers

Figure 3: The search models constructed in this work.

1) Direct arrival model

As shown in Figure 3(a), Ls is the set of routes that pass through the starting point, S, while Le is the set of routes passing through the corresponding set of end points, E. The intersection between Ls and Le is then searched for, and each element in this intersection corresponds to a direct arrival plan.

2) 1-transfer model

As shown in Figure 3(b), Ms is the set of stops that can be reached from S, and Lms is the set of routes that pass through the stops in Ms and their proximal stops. The intersection between Lms and Le is then searched for, and all elements in this intersection correspond to 1-transfer plans. The search is ended when the number of plans that have been found reaches maxCases.

3) 2-transfer model

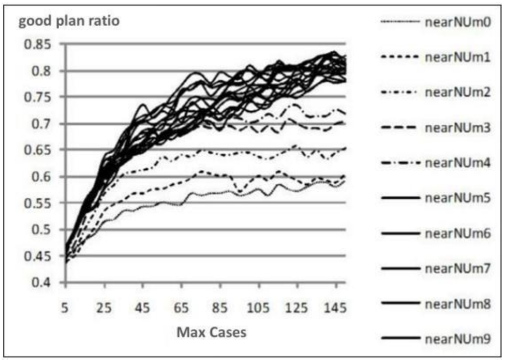
As shown in Figure 3(c), Me is the set of stops that can directly reach the stops of the end points set, E, and Lme is the set of routes that pass through the stops in Me and their proximal stops. The intersection between Lms and Lme is then searched for, and the elements of this intersection correspond to the 2-transfer plans. The search is ended when the number of plans that have been found reaches maxCases.

4) 3-transfer model

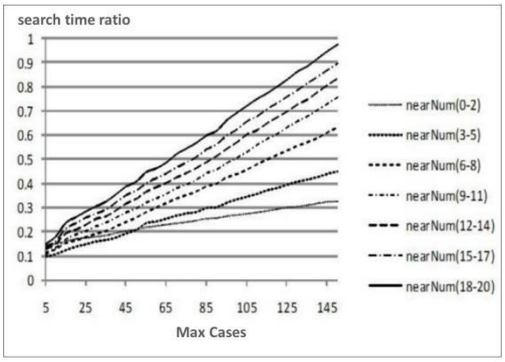
As shown in Figure 3(d), Mms is the set of stops that can be reached from S after a single transfer, and Lm is the set of all routes passing through the stops in Mms and their proximal stops. The intersection between Lm and Lme is then searched for, and the elements in this intersection correspond to the 3-transfer plans. The search is ended when the number of plans that have been found reaches maxCases.

3.3. Optimal threshold values

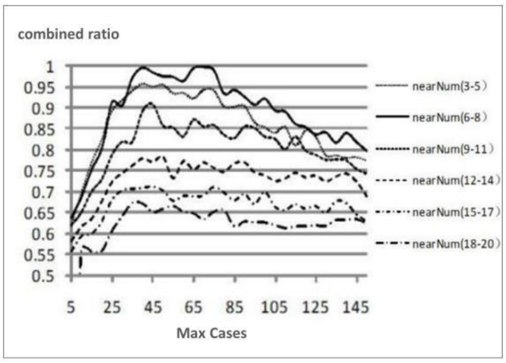
If there are multiple start and end points and walking is taken into account, the nearNum threshold ensures that the optimal plans will be prioritized in the search process, while maxCases limits the search time. To determine the optimal values of nearNum and maxCases, we have randomly selected 200 pairs of stops as samples and varied the values of nearNum and maxCases; the resulting number of good plans and search times were then statistically analyzed. Here, the range of nearNum was nearNum ∈[0,20], and its step length was 1; the range of maxCases was maxCases ∈[5,150], and its step length was 5.



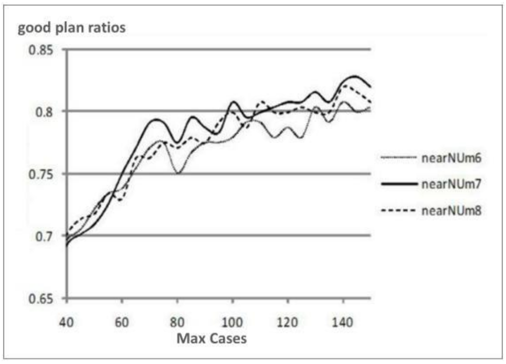
(c) good plan ratios



(b) search time ratios



(c) combined ratios



(d) optimal nearNum value

Figure 4: Experimental Results.

1) The number of good plans

For convenience, we used the proportion of good plans among the plans that have been found (denoted the "good plan ratio") as the basis of this statistical analysis. The average good plan ratios with 200 samples were calculated, as shown in Figure 4(a). Here, it is shown that the good plan ratio increases with the number of cases and ultimately reaches a plateau. Also, the good plan ratio was significantly lower for nearNum values of 0, 1 and 2.

2) Search time

For convenience, we used the ratio between se Figurearch times and the maximum search time (denoted the "search time ratio") as the basis of this analysis. The average calculated values of the search time ratios with 200 samples are shown in Figure 4(b). Here, one can see that the search time ratio increases as maxCases increases, and the rate of increase of the search time ratio increases with nearNum.

3) Combined analysis

To examine the number of good plans and search times in a unified manner, we performed a statistical analysis on the ratio of the square of good plan ratios to the square root of search time ratios, henceforth defined as the "combined ratio". The data corresponding to nearNum values of 0, 1 and 2 were excluded from this analysis, and the average combined ratios of 200 samples were calculated and normalized, as shown in Figure 4(c). It is shown that the combined ratio reaches its optimal value with nearNum values of 6 – 8 and maxCases values of 40 to 70. Based on the previous conclusions, the peak values around maxCases = 40 are preferable in terms of search times, whereas the peak values around maxCases = 70 are preferable in terms of good plan ratios. Since the good plan ratio is more important for this application, we chose maxCases = 70 as the optimal value for maxCases. Figure 4(d) illustrates the good plan ratios with nearNum values of 6, 7 and 8. In this comparison, it is shown that nearNum = 7 yields the highest good plan ratios. However, do note that the optimal values for nearNum and maxCases may depend on the maximum walking distance defined for the algorithm and the scale of public transit in the city.

The effectiveness of our augmented algorithm has been validated through experiments in Changzhi City in the Shanxi Province (64 public transit routes, 905 public transit stops), Weifang City in the Shandong Province (115 public transit routes, 1317 public transit stops), and Langfang City in Hebei Province (44 public transit routes, 669 public transit stops). Here, we used the public transit data of Changzhi City in the Shanxi Province as an example, in which 200 pairs of public transit stops were used as sample data, alongside the optimal threshold values obtained in the previous section. In experiments comparing the optimal routes and query times of the least transfer algorithm before and after the augmentations proposed in this work, we have found that both of these aspects are significantly improved in the augmented algorithm.

4.1. Optimal route

In this example, the "Baihou" stop was set as the starting point, while the "Bayi Hotel" stop was set as the end point. The solid and dotted lines in Figure 5 correspond to the optimal routes calculated by the original least transfer algorithm and the augmented least transfer algorithm, respectively. The route indicated by the solid line is: take Bus No.2 to the Gongdiangongsi stop (3 stops), cross the road to the opposite stop, and take Bus No.23 to the Bayi Hotel stop (6 stops); this takes a total of 30 minutes. The route indicated by the dotted line is: walk 600 meters to the Jiaojingzhidui stop, and take Bus No. 306 or Bus No. 902 for 5 stops to reach the Bayi Hotel stop; this takes a total of 20 minutes. This comparison therefore demonstrates that the optimal route calculated by the augmented algorithm used a reasonable walking distance to replace a bus journey, thus shortening the overall transit process and consuming less time. In practice, this is the route that would be preferred by most people.

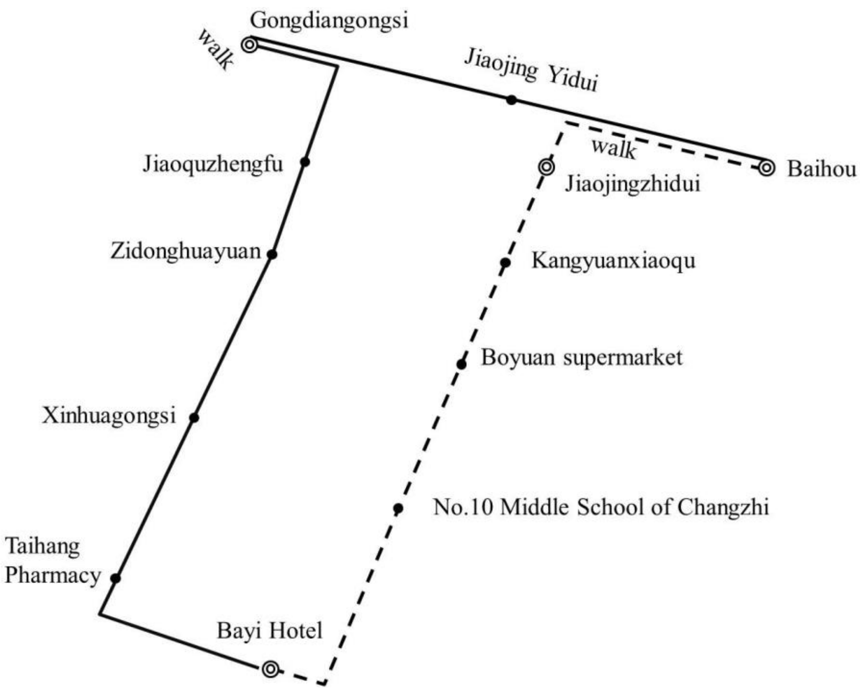


Figure 5: Comparison between optimal routes.

4.2. Query times

The experimentally recorded plan-querying times of the algorithm before and after augmentation are shown in Figure 6. Here, it is shown that the augmented algorithm significantly improves on the original algorithm in terms of query times, thus demonstrating that the setting of threshold values and the use of 1-to-N search models can reduce query times.

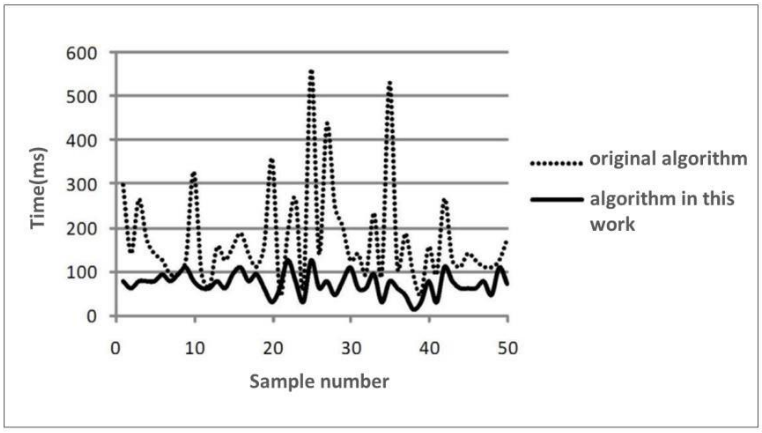


Figure 6: Comparison between the query times of each algorithm

In this work, we have proposed an optimized least transfer algorithm for public transit, and used the 2014 public transit data of Changzhi City in the Shanxi Province to validate our algorithm. The experimental results prove that the dynamic determination of start and end points, incorporation of threshold values and use of the one-to-N search model effectively addresses the efficiency and search time issues of the least transfer algorithm.

Acknowledgements

This research was supported by the National Surveying and Mapping Projects (A1705).

Chang L., Liu R. Y., Zhang F. et al. (2011). The optimal path algorithm of bus network considered multi-mode transferring. Journal of Zhejiang University, 38(6):701-707.

Chen S. P., Tan J. J. (2010). Public transportation transfer based on integration of geometric and semantic rules applied on transportation networks. Science of Surveying & Mapping, 35(3):159-161.

Fu Z., Zhang W., Meng Q. (2010). A Study of Bus Data Model and Transfer Algorithm Based on GIS. Bulletin of Surveying and Mapping, 7:15-18

Liao C., Cai Z., Du Q. et al. (2006). Modern Realization of Public Traffic Optimal Path Based on Least Transfers. Geomatics & Information Science of Wuhan University, 31(10):904-907.

Liu X. (2011). Research and Design of Bus Transfer GIS based on iPhone. Shandong University.

Seoul Development Institute (2005). Changing Profile of Seoul: Major Statistics and Trends. Seoul Development Institute.

Wang X. C., Qing Y. E., Yue K. H., et al. (2011). City Bus with Walking Transfer Algorithm and Application. Journal of Chuxiong Normal University, 26 (3):11-15.