Heuristic Algorithms for the Dynamic Taxipooling Problem Based on Intelligent Transportation System Technologies

Chi-Chung Tao
Tamkang University
cctao@mail.tku.edu.tw

Chun-Ying Chen
National Central University
93342013@cc.ncu.edu.tw

Abstract

The convergence of the Intelligent Transportation System (ITS) technologies has given rise to new opportunities for creative and incentive taxi services such as taxipooling. Taxipooling is similar to carpooling which is based on the idea that sets of users having the same travel destination and sharing vehicles. This paper presents two heuristic algorithms based on greedy method and the time-space network for the case of one origin to many destinations ("one-to-many") and many origins to one destination ("many-to-one"). These algorithms are used to support a field trial at Taipei Nei-Hu Science and Technology Park in Taiwan. The results of numerical tests have demonstrated that the outcomes of these heuristic algorithms are fairly plausible.

1. Introduction

The convergence of the Intelligent Transportation System (ITS) technologies, including the Internet, wireless communications, geographic information system (GIS), positioning technologies, and mobile devices, has given rise to new opportunities for taxi services. With ITS technologies taxi operators can monitor their entire fleet and track mobile user's location and movement on GIS-based maps. Due to any information or message about each taxi or user can be provided visually and interactively, taxi has become a new public transportation system which has high mobility and high accessibility in the same manner as privately owned cars.

According to the statistics report [1], the amount of taxies in Taipei city has reached 32,824 vehicles up to the end of 2006. The average daily working time per taxi driver is 10 hours, but the vacancy time is 3 hours. Most of taxi drivers are willing to take incentive measures such as taxipooling to improve the average level of occupancy. Taxipooling is similar to carpooling which is based on the idea that sets of users having the same travel destination and sharing vehicles.

The carpooling problem has been modeled as a vehicle routing problem with pickup and deliveries and time windows, and solved with heuristic concepts. Due to lack of an efficient information and communication support, the problem of automatically creating dynamic ride match lists upon carpooling or taxipooling demand has been researched for many years. Healy [2], Cordeau [3], Fu [4,5], Attanasio [6], Diana [7], Baldacci et al. [8] and Calvo et al. [9] treated the carpooling problem as a special case of dial-a-ride problem and proposed their solution techniques considering the size and peculiarity of the application faced. Previous work suggests that dynamic ride matching differs from regular carpooling and taxipooling in that ridesharing is arranged for individual trips rather than for trips made on a regular basis and requests for ridesharing can be made close to the time when travel is desired [10].

Dynamic rideshare matching differs from traditional rideshare matching in following ways: Traditional systems assume the traveler has a fixed schedule and a fixed set of origins and destinations [11]. A dynamic system must take into account each trip individually and be able to adjust trips to arbitrary origins and destinations at anytime by matching users' individual trips. The other major difference is that dynamic ride matching systems must offer the real-time match information to the user to accommodate short-term (e.g. same day) travel as well as long-term (e.g. future days or weeks) trips. Using ITS technologies the requirements of dynamic rideshare matching are easier to meet than those of for traditional rideshare applications [12].

This paper is aimed on modeling a dynamic rideshare matching application of taking into account the Internet and wireless communication network infrastructure to meet the requirements of taxi passengers from one origin to many destinations ("one-to-many") and from many origins to one destination

Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)
0-7695-2874-0/07 $25.00 © 2007 IEEE

("many-to-one"). First, the specific taxipooling problem is addressed and the concept of solution procedure is also introduced. Then, solution techniques for two cases of "one-to-many" and "many-to-one" are presented. Finally, computational results are analyzed and the conclusion follows.

2. The problem and algorithms

The problem being considered in this paper is defined around a situation in which a number of potential passengers that daily commute between their house and workplace. These commuters need taxipooling services with the type of "many-to-one" in the morning peak hour and the type of "one-to-many" in the afternoon peak hour. Passengers can access the taxi call center either through a World Wide Web (WWW) browser or via mobile phones at any time. Given this situation, the problem at hand is deciding which passengers should be matched and assigned to a taxi in a way of minimizing the distance traveled and time of both passengers and taxi.

Having reviewed related literature, the greedy heuristic method is chosen as the algorithm in this paper. The solution procedure is to search the nearest passenger until satisfying the constraints of passenger's preference and vehicle capacity. The algorithm descriptions for the type of "one-to-many" and the type of "many-to-one" are given respectively as follows.

2.1. Algorithm procedure

Although the capacity constraint is set to be four passengers for the taxipooling service, it is also suitable for over four persons if the pooling vehicle larger than a car. The algorithm procedure is shown in Fig. 1.

Step0: Setting parameter D and T.

1. D is used to restrict the searchable scope that prevents routing distance from being too long.
2. T means the maximum passenger waiting time.

Step1:Sorting by taxipooling passengers' preferences which can be classified into following 9 types:

1. Acceptable number of taxipooling passenger is two and female only.
2. Acceptable number of taxipooling passenger is two and male only.
3. Acceptable number of taxipooling passenger for is two and no request for male or female preference.
4. Acceptable number of taxipooling passenger is three and female only.
5. Acceptable number of taxipooling passenger is three and male only.
6. Acceptable number of taxipooling passenger is three and no request for male or female preference.
7. Only female taxipooling passengers are acceptable.
8. Only male taxipooling passengers are acceptable.
9. No request.

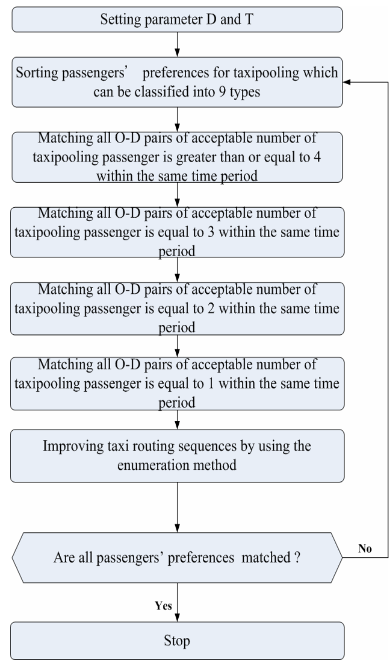

Figure 1. The algorithm procedure

The matching process will continue from type 1 to type 9 till the major candidate passengers are matched by considering the strictest constraint of preference with the first priority. If they can not be matched, step 2 will not be taken.

Step2: Matching all O-D pairs of acceptable number of taxipooling passenger is greater than or equal to four within the same time period. A simple example is shown as follows:

6:30	6:35	6:40
O→D1 ~~5~~ 1	O→D1 2	....
O→D2 1	O→D2 ~~5~~ 1	....
O→D3 3	O→D3 ~~5~~ 1	....
O→D4 ~~8~~ 0	O→D4 1	....
O→D5 1	O→D5 ~~7~~ 3	....
O→D6 1	O→D6 1	....
O→D7 1	O→D7 1	....

Step3: Matching all O-D pairs of acceptable number of taxipooling passenger is equal to three within the same time period.

Step4: Matching all O-D pairs of acceptable number of taxipooling passenger is equal to two within the same time period.

0-7695-2874-0/07 $25.00

Step5: Matching all O-D pairs of acceptable number of passenger for taxipooling is equal to one within the same time period.

Step6: Improving taxi routing sequences by using the enumeration method. For example, if the original route is O→D1→D2→D3, all possible sequences of O→D1→D3→D2,O→D2→D1→D3,O→D2→D3→D1,O→D3→D1→D2, and O→D3→D2→D1 will be searched to identify the shortest route.

Step7: If all passengers' preferences are matched, then stop; else, go to step 1.

2.2. The one-to-many algorithm

The one-to-many algorithm is based on a time-space network shown in Fig. 2. The vertical axis represents the time duration and the horizontal axis indicates the passenger location. A node stands for a location at a specific time and an arc designates an activity for a taxi. Theoretically, the higher the density of the nodes is, the larger the problem size will be. Therefore, a suitable density of nodes should be selected according to actual requirements. In this paper the interval between the two nodes is 15 minutes. Two types of arcs are described as follows:

1. Service arc
   A service arc, marked by (1) in Fig. 2, represents a taxi movement between two different locations. All possible service arcs between two continuous locations within a reasonable time period are plotted in the network.
2. Holding arc
   The holding arc, marked by (2) in Fig. 2, represents the holding of a taxi at a location within a time period. The algorithm allows taxi waiting at origin in order to increase the matching probability.

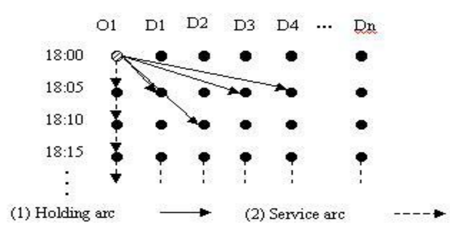

Figure 2. The time-space network of "one-to-many"

The searching method is shown in Fig. 3. The mark "o" represents an origin, while the mark "d" stands for a destination. The first step is to set a passenger's "o" and begin to expand searching areas until the nearest passenger's "d" is found with the constraints of D, T and passengers' preferences. The flowchart of the matching process for the type of "one-to-many" is shown in Fig. 4.

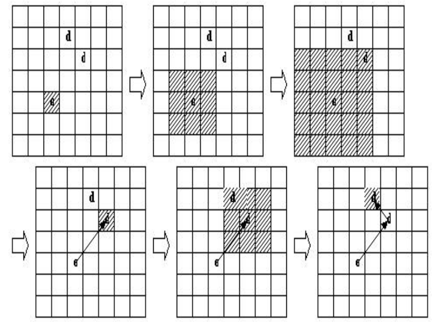

Figure 3. The searching method on space locations

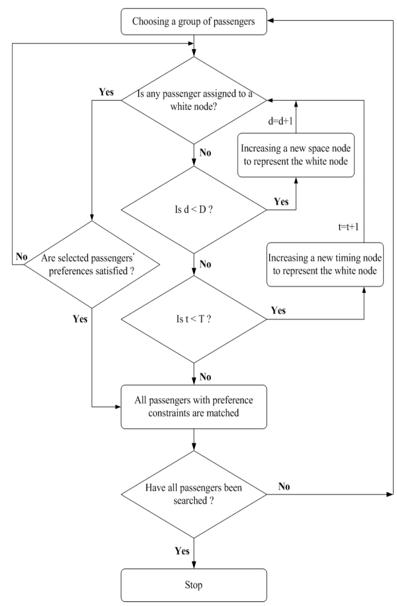

Figure 4. Flowchart of matching process for "one-to-many"

The detailed steps of the matching process are illustrated as follows:

Step1: A group of passengers is chosen as (O1, 18:00) in Fig. 5.

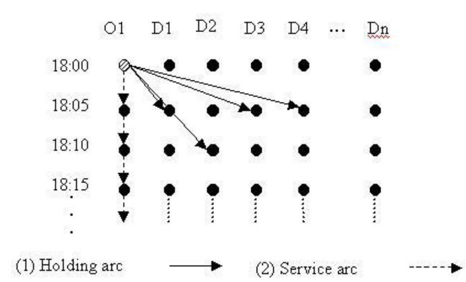

Figure 5. Step 1 of matching process for "one-to-many"

Step2: If any taxipooling passenger is assigned to a white node shown in Fig. 6, then go to step 3; else, go to step 4.

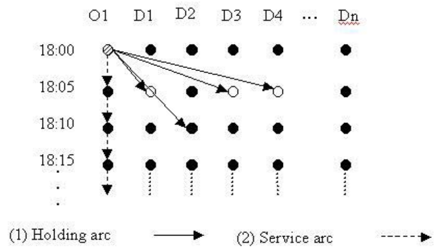

Figure 6. Step 2 of matching process for "one-to-many"

Step3: Passengers are chosen and the constraints of passengers' preferences are also checked. If the constraints are not satisfied, go to step 8; else, a taxi is assigned to the node (D1, 18:05) shown in Fig. 7 and then go to step2.

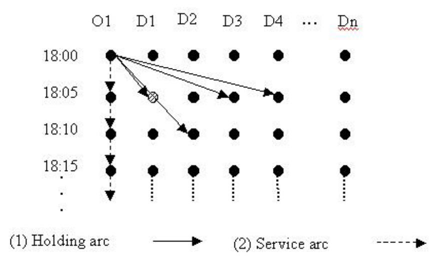

Figure 7. Step 3 of matching process for "one-to-many"

Step4: If the searchable area is smaller than parameter D, then go to step 5; else, go to step 6.

Step5: A new space node is increased to represent the white node shown in Fig. 8.

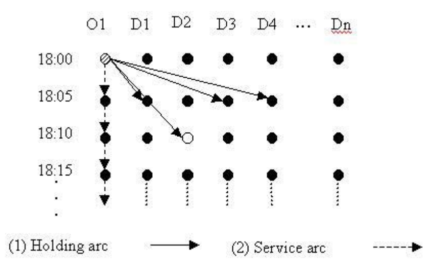

Figure 8. Step 5 of matching process for "one-to-many"

Step6: If the searchable area is smaller than parameter T, then go to step 7; else, go to step 8.

Step7: A new timing node is increased to represent the white node shown in Fig. 9.

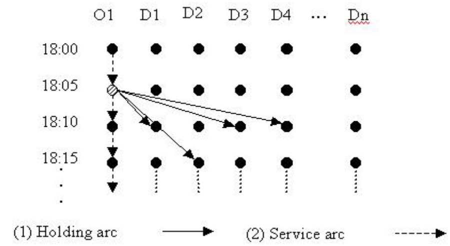

Figure 9. Step 7 of matching process for "one-to-many"

Step8: All select passengers with preference constraints are matched.

Step9: If all passengers at location O1 have been searched, then stop; else, go to step 1.

2.3. The many-to-one algorithm

The horizontal and vertical axes are defined to be the same as those in the "one-to-many" network. The significant difference is the O-D pairs of the type of "many-to-one" that stands for multiple origins and only one destination shown in Fig. 10. In this case, holding arcs indicate passengers who are waiting for the next taxipooling passengers. In practice, if a taxi arrives early, the driver will contact the call center to remind passengers that they are expected to arrive on time.

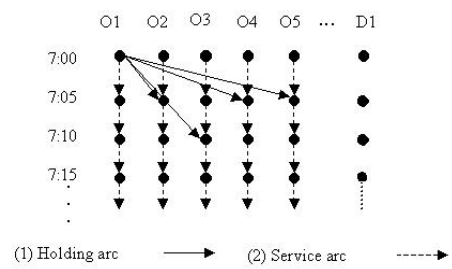

Figure 10. The time-space network of "many-to-one"

The steps of matching process for "many-to-one" are similar to those of "one-to-many". The major differences are step 1, step 7 and step 9 which can be illustrated as follows:

Step1: A group of passengers is chosen randomly from all origins.

Step7: A new timing node is increased to represent the white node shown in figure 11. For example, suppose a taxipooling passenger has been searched at (O1, 7:00). Then (O2, 7:05), (O3, 7:10), (O4, 7:05) and (O5, 7:05) and his constraints of preferences are also checked. If there is no further passenger, the white nodes in figure 11 are checked. It is notified that the white node (O1, 7:05) for the same location should be

Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)
0-7695-2874-0/07 $25.00 © 2007 IEEE

first checked because the other white nodes might be influenced by traffic jams.

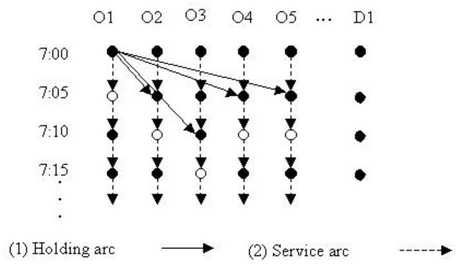

Figure 11. Step 7 of matching process for "many-to-one"

Step9: If all passenger locations (O1, O2, O3, …) are searched, then stop; else, go to step 1.

3. Numerical tests

To test how well the algorithms may be applied in the real world, a field trial of taxipooling is conducted at Taipei Nei-Hu Science and Technology Park in Taiwan from October 26 to November 17 in 2006[13]. There are 10 taxis and 798 passengers participating in this pilot project. The "one-to-many" algorithm is performed from 6:00 AM to 9:00 AM. The "many-to-one" algorithm is performed from 6:00PM to 9:00 PM. The physical system architecture of the taxipooling service can be shown in Fig. 12.

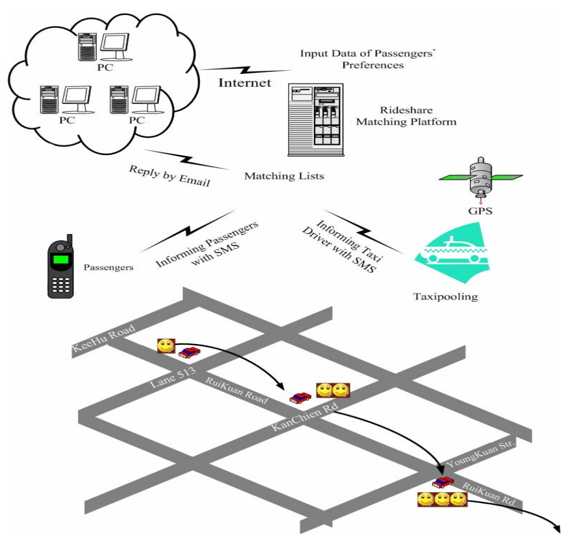

Figure 12. The physical system architecture of the taxipooling service

As shown in Tab. 1, matching success rates for "many-to-one" and "one-to-many" are 53.9% and 53.6% respectively. There are 481 passengers have been matched in total, the matching success rate is 60.3% on the whole. In additional, the computation time of each test is less than 1 second. It is, of course, to be expected that the matching success rate is zero if number of passengers is too small or passenger locations are too dispersive. In Tab. 1 the matching success rate of the 2nd test in the afternoon, the 8th test all day and the 9th test in the morning are zero.

Table 1. Matching success rates of the field trial in Taipei

Test	many-to-one (morning)			one-to-many (afternoon)		
	# join matching	# matching success	success rate	# join matching	# matching success	success rate
1st	-	-	-	15	9	60.0%
2nd	13	8	61.5%	3	0	0.0%
3rd	8	5	62.5%	12	7	58.3%
4th	19	13	68.4%	17	10	58.8%
5th	30	18	60.0%	35	22	62.9%
6th	25	14	56.0%	45	27	60.0%
7th	48	30	62.5%	38	24	63.2%
8th	3	0	0.0%	2	0	0.0%
9th	2	0	0.0%	42	24	57.1%
10th	36	23	63.9%	9	5	55.6%
11th	51	29	56.9%	40	23	57.5%
12th	31	20	64.5%	30	19	63.3%
13th	27	18	66.7%	26	16	61.5%
14th	20	12	60.0%	29	18	62.1%
15th	26	14	53.9%	17	11	64.7%
16th	20	13	65.0%	16	10	62.5%
17th	38	23	60.5%	25	16	64.0%
Average	23.35	14.12	53.9%	23.59	14.18	53.6%

As shown in Fig. 13, the average number of passenger per taxi (e.g. load factor) for "many-to-one" and "one-to-many" is 2.4 and 2.3 respectively. There is no significant difference in load factor between "many-to-one" and "one-to-many". The load factor will decrease if the demand for taxipooling declines.

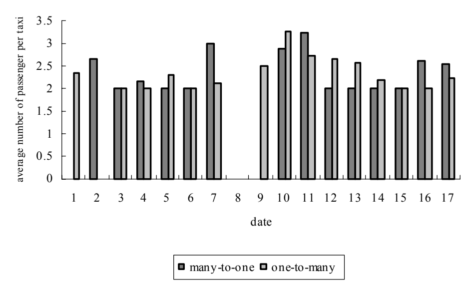

Figure 13. Average number of passenger per taxi of the field trial in Taipei

The average saving distance is 58,215 km for "many-to-one" and 62,272 km for "one-to-many" which are shown in Fig. 14 and Fig. 15 respectively. The factors affect the average saving distance including matching success rates, distributions of passenger locations, taxi routing sequences, etc. The results reveal that a good dynamic matching algorithm will save more travel distances for the taxipooling problem.

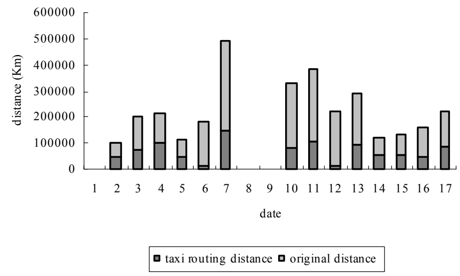

Figure 14. Average saving distance for "many-to-one"

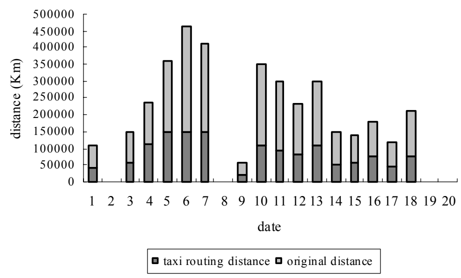

Figure 15. Average saving distance for "one-to-many"

4. Conclusions

Two heuristic algorithms based on greedy method and the time-space network are developed to efficiently solve the dynamic taxipooling problem for 'one-to-many" and "many-to-one" respectively. These algorithms are used to support a field trial at Taipei Nei-Hu Science and Technology Park in Taiwan. The results of numerical tests demonstrated that the outcomes of these heuristic algorithms are fairly plausible. The average matching success rate is 60.3% and the average saving distance is 63873 km on the whole.

However, the developed algorithms are applicable to the case of "one-to-many" and "many-to-one" which can more or less describe the commuter travel behavior for certain urban areas. The case of "many-to-many" which fully represents dynamic ride matching with any O-D pairs for taxipooling problem is under development. Some advanced heuristic techniques such as tabu search method, threshold accepting method, genetic algorithm, lagrangian relaxation or column generation may provide good solutions if the proposed heuristic algorithms are not able to cope with the large scale time-space network problems. This could be a direction of future work.

5. References

[1] Ministry of Transportation and Communications, Annual Statistical Report 2006 (in Chinese), Taiwan, 2006.

[2] P. Healy, R. Moll, A New Extension of Local Search Applied to the Dial-A-Ride Problem, European Journal of Operational Research, Vol.83, pp.83-104, 1995.

[3] J. F. Cordeau, G. Laporte, A Tabu Search Heuristic for the Static Multi-vehicle Dial-a-ride Problem, Transportation Research Part B, Vol.37, pp.579–594, 2001.

[4] L. Fu, A Simulation Model for Evaluating Advanced Dial-a-ride Paratransit Systems, Transportation Research, Vol. 36A, pp.291-307, 2002.

[5] L. Fu, Scheduling Dial-a-ride Paratransit under Time-varying Stochastic Congestion, Transportation Research, Vol. 36B, pp. 485-506, 2002.

[6] A. Attanasio, J. F. Cordeau, G. Ghiani, G. Laporte, Parallel Tabu Search Heuristics for the Dynamic Multi-vehicle Dial-a-ride Problem, Parallel Computing, Vol.30, pp. 377-387, 2004.

[7] M. Diana, M. Maged, Dessouky, A New Regret Insertion Heuristic for Solving Large-scale Dial-a-ride Problems with Time Windows, Transportation Research Part B, Vol.38, pp.539–p.557, 2004.

[8] R. Baldacci, V. Maniezzo, A. Mingozzi, An Exact Method for the Carpooling Problem Based on Lagrangean Column Generation, Operations Research, Vol. 52, No. 3, pp. 422–439, 2004.

[9] R. W. Calvo, F. L. Luigi, P. Haastrup, V. Maniezzo, A Distributed Geographic Information System for the Daily Carpooling Problem, Computers & Operations Research, Vol.31, pp. 2263–2278, 2004.

[10] R. F. Casey, L. N. Labell, R. Holmstrom, J. A. LoVecchio, C. L. Schweiger, T. Sheehan, Transportation Demand, Management, Technology, Chap. 5 Advanced Public Transportation Systems: The State of the Art, Update Report No. FTA-MA-26-707-96-1, US DOT, FTA, pp. 109-139, 1996.

[11] S. Michalak, J. Spyridakis, M. Haselkorn, B. Goble, C. Blumenthal, Assessing Users' Needs for Dynamic Ridesharing, Transportation Research Record 1459, TRB, National Research Council, Washington, DC, pp. 32-38, 1994.

[12] D. J. Daily, D. Loseff, D. Meyers, Seattle Smart Traveler: Dynamic Ridematching on the WWW, Transportation Research Part C, Vol.7, pp.17-32, 1999.

[13] C. C. Tao, C. Y. Chen, C. J. Wu, W. H. Lee, B. K. Chang, C.C. Hung, C.A. Hsu, Deployment of High Occupancy Vehicle Project using ITS Technologies-A Demonstration on Ride Sharing System in Urban Areas – Phase II (in Chinese), Final Report, Ministry of Transportation and Communications, Taiwan, 2006.