



A novel approach to independent taxi scheduling problem based on stable matching

Ruibin Bai^{1*}, Jiawei Li², Jason AD Atkin² and Graham Kendall^{2,3}

¹University of Nottingham Ningbo, Ningbo, China; ²University of Nottingham, Nottingham, UK; and

³University of Nottingham Malaysia Campus, Malaysia

This paper describes a taxi scheduling system, which aims to improve the overall efficiency of the system, both from the perspective of the drivers and the customers. This is of particular relevance to Chinese cities, where hailing a taxi on the street is by far the most common way in which taxis are requested, since the majority of taxi drivers operate independently, rather than working for a company. The mobile phone and Global Positioning System-based taxi scheduling system, which is described in this paper, aims to provide a decision support system for taxi drivers and facilitates direct information exchange between taxi drivers and passengers, while allowing drivers to remain independent. The taxi scheduling problem is considered to be a non-cooperative game between taxi drivers and a description of this problem is given. We adopt an efficient algorithm to discover a Nash equilibrium, such that each taxi driver and passenger cannot benefit from changing their assigned partner. Two computational examples are given to illustrate the effectiveness of the approach.

Journal of the Operational Research Society (2014) **65**(10), 1501–1510. doi:10.1057/jors.2013.96

Published online 4 September 2013

Keywords: taxi scheduling; game theory; stable marriage problem; GPS; street hailing

1. Introduction

Taxis play an important role in modern public transportation networks, especially in countries such as China and India, where bus and underground train services are still under development and do not currently provide the coverage required by passengers. Taking a taxi is a convenient way to get to your destination, although it is usually slightly more expensive than buses and subways. According to a 2006 survey, there are approximately 1 million taxis in service around mainland China and the 2 million people who are involved in the taxi industry generate an annual turnover of more than 29.5 billion US dollars (National Bureau of Statistics of China, 2006).

We can broadly classify taxi services into three categories: street hailing, taxi-stands, and pre-arranged bookings (Li, 2006, pp 25–27). Street hailing is a common method for getting a taxi, especially in small cities where taxi services are on a relatively small scale. Street hailing is simple and has no requirement for expensive equipment or systems. However, as will be analysed later in this section, street hailing causes several problems, including low taxi utilization, long passenger waiting times, road safety issues, and traffic congestion. Taxi-stands have fewer problems but can only serve a small group of passengers at fixed locations, for example, city centres, bus/train stations, airports, etc. Pre-arranged booking is widely considered to be a more advanced and efficient taxi dispatching method, but is

impractical when there are many independent taxi drivers. The booking can be performed via traditional telephone calls or by utilizing other communication technologies (such as websites or text messages). A taxi dispatching centre is often employed to collect all the passenger information and coordinate the taxi dispatching centrally. The dispatching systems often utilize Global Positioning System (GPS) devices and wireless communication equipment. The advantage of such a pre-booking system is that the demands of passengers are clear and known in advance. One drawback is that the system is often expensive and therefore only suitable for large taxi companies. In addition, the system relies upon the mutual trust between taxi drivers and passengers, in that neither party will break the agreed appointment. Unfortunately, in reality, the breaking of appointments happens frequently in many countries.

Unlike in Europe and the USA, pre-arranging bookings is not a popular method for hailing taxis in China. Most taxi drivers are self-employed, although they may be registered as employees for some taxi companies. It is hard to build a pre-arranged booking mechanism where taxi drivers who are self-employed (or employed by many different taxi companies) would work cooperatively. An efficient communication network would also be needed to implement this type of taxi scheduling, especially in a city with a large population. In China, a very limited number of taxi companies provide a pre-arranged booking service, except in some of the larger cities such as Beijing and Shanghai. Therefore, street hailing is still the most frequent method of getting a taxi in China and approximately 85% of taxi services are operated in this way. However, as

*Correspondence: Ruibin Bai, School of Computer Science, University of Nottingham Ningbo, Ningbo, 315100, China.
E-mail: ruibin.bai@nottingham.ac.uk

mentioned earlier, street hailing is inefficient and poses both safety and congestion problems. Taxi drivers are unable to predict exactly when and where to meet the next passenger and passengers cannot be sure how long they will have to wait for the arrival of a taxi, leading to high mileage waste for taxi drivers and longer waiting times for passengers. In addition, taxi drivers need to concentrate on both driving and finding the next passenger, increasing the risk of accidents. Frequently, too many taxis crowd the city centres and passenger hotspots, leading to congestion.

There have been several research projects and technological innovations for pre-arranged booking systems, often abstracted and formulated as Dial-A-Ride Problem (DARP), which falls into the broader category of classic Vehicle Routing Problems, or the Traveling Repair Problem (Cordeau and Laporte, 2007). Several on-line algorithms have been developed, using linear programming (Deng *et al.*, 1992), heuristics (Madsen *et al.*, 1995; Irani *et al.*, 2004; Luo and Schonfeld, 2007) and meta-heuristics (Nkoane, 2009). GPS and Geographic Information Systems (GIS) have been used by vehicle dispatching systems in many cities, to improve the efficiency of their taxi services (Lee *et al.*, 2003; Liao, 2003; Prakash and Kulkarni, 2003; Glaschenko *et al.*, 2009). For example, a GPS-based Automatic Vehicle Location and Dispatching System greatly improved the quality and efficiency of taxi services in Singapore. Once a request for a taxi is received, the control centre will send signals to the taxis that are located within 10 km of the passenger. If one of the drivers accepts the job, he/she can respond to the control centre by merely pressing a button that is installed in his/her taxi. Once the job has been assigned, information about the taxi number and expected time of arrival will be immediately sent to the passenger (Lee *et al.*, 2003; Liao, 2003).

Research and innovation to improve the street hailing method has been very limited. The most relevant research, that we are aware of, includes models of the taxi market to analyse the demand and supply equilibrium (Cairns and Liston-Heyes, 1996; Flores-Guri, 2003), the bilateral searching and meeting between passengers and taxis on a road network (Yang *et al.*, 2010b), search friction between taxi drivers and passengers (Yang and Yang, 2011), and the taxi cruising policy, to improve the efficiency of the taxi service (Flores-Guri, 2005). However, models that consider scheduling algorithms for street hailing are currently lacking. Our proposed system is a practical, low cost extension of the street hailing system. It helps handle the important problem of search friction, where the imperfection of information exchange can reduce the quality of the result and will not inhibit independent taxi drivers from strategically positioning themselves when they do not currently have passengers.

The lack of research in this field is probably due to the fact that most taxis that are involved in street hailing are self-employed and independent. Unlike in a centralized system, coordination and cooperation between these taxis is difficult, or even impossible, without an appropriate mechanism. The

systems and dispatching methods that are used in the pre-arranged booking method are not suitable for street hailing. The independent nature of the taxi drivers using such a system cannot be ignored, and perhaps the most important consideration for such a system is ensuring that it allocates passengers to taxis so that it is always in the best interests of the taxi drivers to keep the passengers that they have been allocated to. The focus of this paper is upon a novel system for extending the street hailing problem, in order to complement existing systems.

In this paper, the problem of assigning independent taxis to customers is considered as a non-cooperative game between taxi drivers. Game theorists have developed Nash equilibrium as the theoretical solution to non-cooperative games (Nash, 1951). If each agent has chosen a strategy and no agent can benefit by changing their strategies while the other players keep theirs unchanged, then the current set of strategy choices and the corresponding payoffs is a Nash equilibrium. A Nash equilibrium in the street hailing problem is a stable matching between taxi drivers and passengers such that no taxi driver can be better off by changing their chosen passenger. There has previously been some applications of equilibrium analysis in logistic and scheduling (Ishii *et al.*, 2013; Kendall and Li, 2013).

In game theory, the Price of Anarchy (PoA) is used to describe the ratio between the payoff from an equilibrium and the maximum social payoff (Corbo and Parkes, 2005). PoA can be a criterion in choosing between multiple equilibria. The stable matching with the highest PoA is always preferred in taxi scheduling.

In the remaining sections, we first describe the street hailing problem. Section 3 introduces a novel game theoretic model of street hailing, along with a solution methodology for the problem. We then adopt the proposed approach and provide two computational examples in Section 4. Section 5 discusses practical implementation issues. Conclusions are presented, along with directions for future work in Section 6.

2. Street hailing using a new system

We propose the following system to improve the taxi street hailing process. The system adopts the latest GPS, GIS, and 3G/GPRS wireless network. It comprises a central server and independent client applications for the passengers and taxis (see Figure 1). The server and clients are connected via a 3G/GPRS network. In our implementation, the clients are smart phones with GPS and GIS support and automatically send location and status (occupied or vacant) information to the server via the 3G/GPRS network. The server periodically analyses the geographical distribution of clients on a digital map, generates a stable taxi-passenger matching (see Section 3), and then broadcasts the allocations back to clients.

In practice, there may be several ways to carry out the taxi scheduling under this proposed system. A straightforward approach would be to provide passengers and taxis with a two-way communication channel to negotiate a manual schedule,

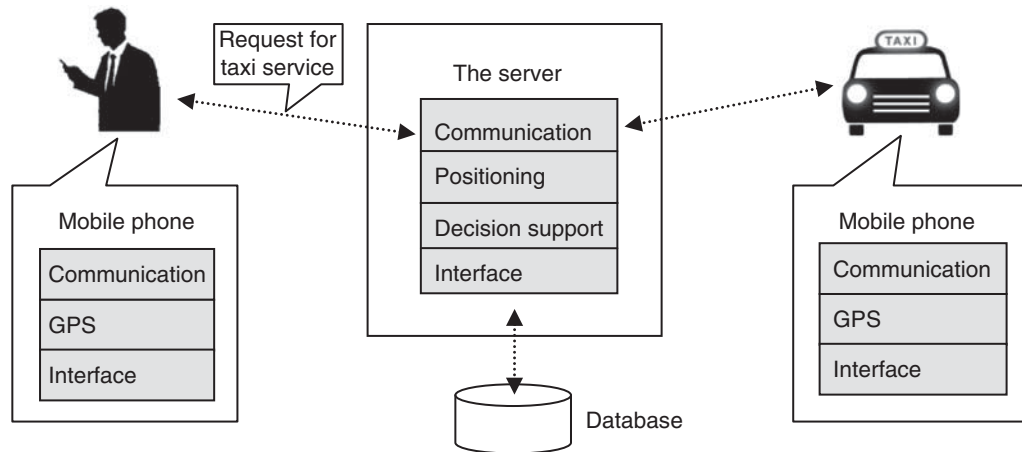


Figure 1 Taxi scheduling system.

although this is probably inefficient. An improved design would provide automatic decision support for the taxis and passengers by taking full advantage of the geographical distribution information of the passengers and taxis on a digital map. This paper focuses on the latter implementation and proposes a scheduling procedure that is illustrated in Figure 2. As can be seen, the city map is divided into smaller zones to facilitate better taxi management and unnecessary complexity. A schedule is generated periodically (defined by the parameter t_w) rather than in real-time. To avoid confusion, the following assumptions are also made about the problem under consideration in this paper:

1. Taxis are independent and compete against each other in order to gain maximum personal return (ie, there is no subset of taxis that plays a cooperative game).
2. A taxi's primary goal is to find the next passenger within the shortest possible time.
3. Scheduling recommendations by the system are only supportive, not compulsory. Therefore, a taxi can decline a recommendation by the system in certain circumstances (eg, if the recommended passenger is too far away, although the division of the city into zones should reduce these occurrences).
4. Locations of passengers and taxis are automatically retrieved from the embedded GPS system and relayed to the server by the client software.
5. A passenger's destination information is not known when a schedule is generated. Therefore, this system cannot plan a taxi's route beyond its very next service. This means the system is not a pre-booking taxi system but serves as an additional system.

We note that some of these assumptions may be relaxed or extended in practical implementations (see the discussions in Section 5). The main purpose of their inclusion in this paper is to introduce a basic problem that can be initially studied to gain some insightful knowledge, and which can be extended later. We believe that the research results reported in this paper, as well

as the new problems presented in this paper, would be of interest for other researchers and is worthy of future research attention.

3. A game theoretic model

In contrast to the previously presented algorithms for the DARP, we develop a game theoretic model that treats the taxi scheduling problem as a non-cooperative game between taxi drivers. The taxi drivers are assumed to be self-interested and compete, rather than cooperate, with each other in serving passengers. Given the locations of passengers and other taxis, each taxi driver will have preferences for some passengers over others and each passenger may also have preferences for taxis (eg, a preference to be picked up as quickly as possible). We aim to provide an algorithm to generate a matching of taxis and passengers that will be accepted by all taxis and passengers. A matching of taxis and passengers is a Nash equilibrium if no taxi or passenger can find a better partner than their current one, so that nobody has any incentive to deviate from the suggested allocation. Such an allocation would then be appropriate for utilization in a system to support street hailing.

Let T_i ($i = 1, \dots, n$) denote the taxis, P_j ($j = 1, \dots, m$) the passengers and t_{ij} the expected time for taxi T_i to reach passenger P_j . Let E_{ij} represent the preference of taxi driver T_i for passenger P_j . We consider a computerized street-hailing system in this paper, rather than a pre-booking system, and we assume that the taxi drivers are not given information about the destination for any passenger. Taxi drivers can, therefore, only utilize the t_{ij} information for their passenger preferences and will prefer passengers who are closer to them. We therefore assume that E_{ij} is entirely determined by t_{ij} and is negatively correlated with t_{ij} . For the experiments in this paper, we assume that E_{ij} is defined as follows, in order to meet these requirements:

$$E_{ij} = 1/t_{ij}, \quad i = 1, \dots, n; j = 1, \dots, m \quad (1)$$

The ability of taxi A to take a passenger off another taxi B to which it has been assigned relies upon both the passenger being

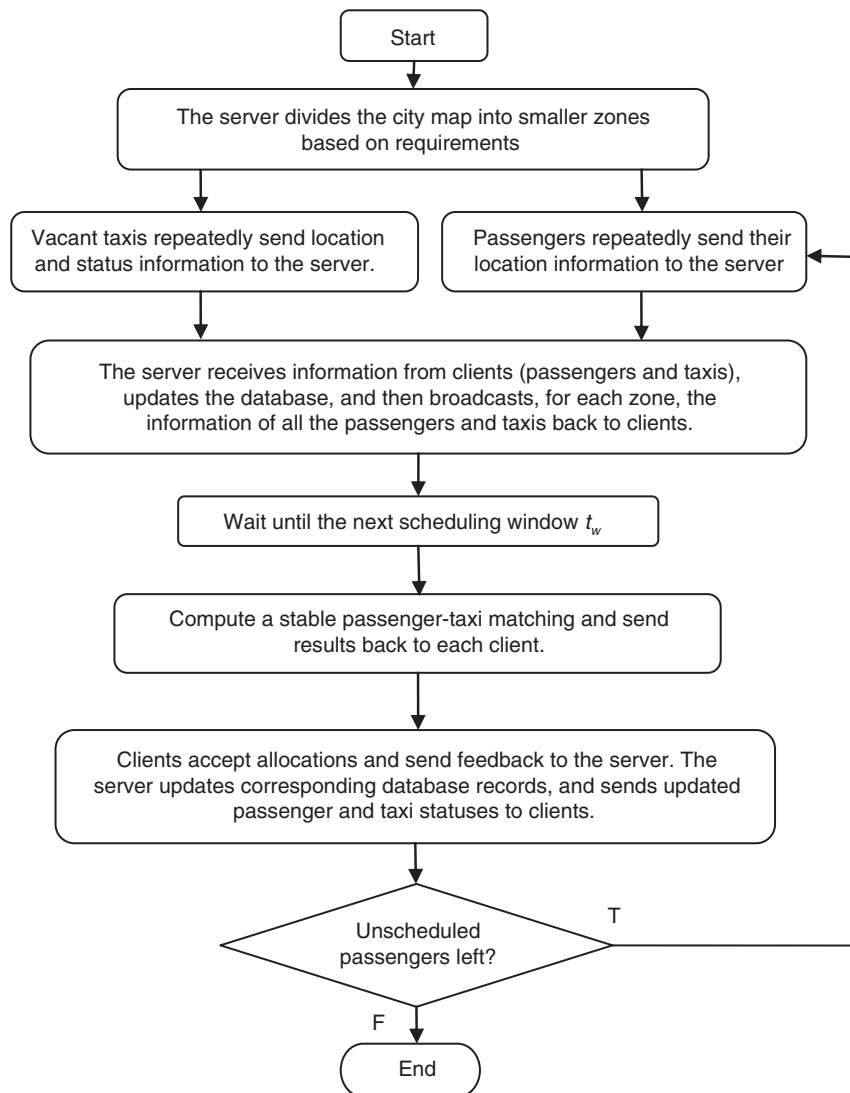


Figure 2 Scheduling algorithm.

willing and taxi A being able to reach the passenger before taxi B . We assume that passengers have no preference for taxis other than that they wish to be picked up as early as possible, so these two factors can be considered to be identical in this case. A passenger's preference, representing the ability of a taxi driver to take this passenger from another taxi driver, could then be determined by assuming that the preference of passenger P_j for taxi T_i is negatively correlated with t_{ij} . Taxi A can then take passenger P_j away from another taxi B if and only if $t_{Aj} < t_{Bj}$, that is, if it can reach the passenger first.

If we consider the taxi drivers' preferences (E_{ij}) as the scoring of one partner and the aforementioned passenger preference as the scoring of the other partner then the taxi-passenger pairing problem can be considered to be a variant of the stable marriage problem. The stable marriage problem (Gusfield and Irving, 1989; Halldorsson *et al*, 2003) involves n men and n women, each of whom has ranked all members of the opposite sex

with a unique number between 1 and n , in order of preference. The objective is to find a set of man-woman pairs such that there are no two people of the opposite sex who would both rather have each other than their current partners. The Gale-Shapley algorithm (Gale and Shapley, 1962) was developed to solve this problem in polynomial time. This algorithm involves a number of 'rounds'. In each round, each unengaged man proposes to the most preferred woman to whom he has not yet proposed. Each woman then considers all proposals to her and chooses the one she most prefers as her partner and becomes engaged to him, potentially 'trading up' from an existing engagement. This process continues until there is no unengaged man. The Gale-Shapley algorithm guarantees a stable matching.

The Gale-Shapley algorithm is still applicable when there are an unequal number of partners, that is, $n \neq m$, so can be applied to the taxi-passenger matching problem. The algorithm is either man-oriented or woman-oriented. That is, each man (or woman)

is paired with the best partner that they can have in any stable matching. In the case of $n \neq m$, a stable matching is guaranteed, but some men (or women) will not be paired.

For the taxi scheduling problem considered here, given the definition of $\{E_{ij}\}$ and the consequent correlation between the two rankings, the Gale–Shapley algorithm can be simplified. We have adopted an algorithm that first sorts all elements in the preference matrix $\{E_{ij}\}$ into decreasing order, then considers each item in turn, from the first to the last. For each element E_{ij} , if neither taxi i nor passenger j has been allocated then i is allocated to j and (i, j) becomes part of the stable matching. The time complexity of the proposed algorithm is $O(mn \times \log(mn))$, since sorting is the slowest element. It is easy to verify that the stable matching is a Nash equilibrium, since the only passengers that a taxi driver could beneficially switch to will have been allocated to a taxi which is even closer to them.

There must be at least one stable matching (Nash equilibrium), given the preference matrix $\{E_{ij}\}$, as long as all $\{E_{ij}\}$ are independent. The proposed algorithm will find one such matching. However, even when stable matchings can be guaranteed, the algorithm will find only one such matching at a time. When there are multiple closest passengers that have the same travel time from a taxi, the algorithm has to make a decision about which passenger to allocate the taxi to. Different combinations of choices between identical travel times will result in different matchings. Therefore, there may exist a large number of stable matches for a problem instance, which has many taxis with identical travel times to passengers. Figure 3(a) provides an example of such a case for the two-taxi, two-passenger problem. In this example, taxi T_1 can be allocated to either passenger P_1 or P_2 and a stable matching will result in either case. However, a lower total cost allocation will be when taxi T_1 is allocated to passenger P_2 than when

taxi T_1 is allocated to passenger P_1 , due to the different travel times for taxi T_2 .

Among all taxi-passenger matchings, the matching with the highest PoA can be attractive because it maximizes the social payoff. However, this optimal (lowest total cost/waiting time) matching may not be stable. For example, consider the two-taxi, two-passenger problem in Figure 3(b). In this case, the optimal matching is to match taxi T_1 with passenger P_2 and taxi T_2 with passenger P_1 , but this is not a stable matching since taxi T_1 could switch to the closer passenger P_1 and reduce its own cost. Since we consider this to be a non-cooperative game, only stable matchings are valid solutions and we only actually care about the differences in the PoA between the stable matchings, and particularly between the current and optimal stable matchings.

Although finding the optimal stable matching could be prohibitively difficult, we have found that (in our experiments) the differences between the total preference values for the stable matchings are actually very small when the number of taxis and passengers are large, as we will show in Section 4.1.

4. Computational examples

In this section, we simulate the taxi scheduling problem in a square area of 50 km × 50 km. Taxis and passengers are most likely to appear around hotspots such as business centres, airports, train stations etc. The locations of taxis and passengers are approximately distributed around these hotspots according to the Poisson distribution (Bailey and Clark, 1987; Li, 2006). In our simulations, the distance between a taxi and a passenger was determined by the Euclidean distance between their locations. Distances were considered to within an accuracy of 0.15 km, so that two distances were considered to be same if their difference was less than 0.15 km.

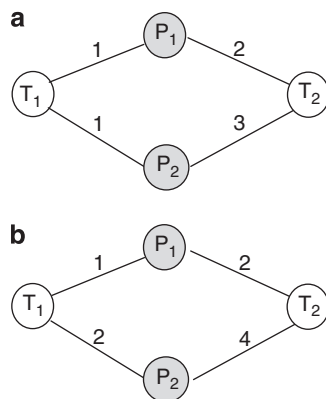


Figure 3 (a) Example of a problem with multiple stable matchings. Regardless of which passenger (P_1 or P_2) the first taxi (T_1) is allocated to, T_2 will be allocated to the other passenger and the matching will be stable. (b) Example of a problem where the optimal social solution is not a stable matching. The optimal social solution has taxi T_1 servicing passenger P_2 and taxi T_2 servicing passenger P_1 , with a total time of 4 units. However, this solution is not stable since taxi T_1 could switch to passenger P_1 , to the benefit of both passenger and taxi driver.

4.1. Optimal stable matching

In this section we describe the simulation that we carried out to compare the stable matching obtained by the simplified Gale–Shapley algorithm to the optimal stable matching, which is the stable matching with minimum total pickup distance. In the simulation, a taxi scheduling problem with 1000 taxis and 1000 passengers is solved using the simplified Gale–Shapley algorithm described earlier. The location of taxis and passengers is shown in Figure 4. There are three hotspots in this simulated service area. The area is divided into 333×333 subareas, each of which is a 0.15 km × 0.15 km square. All positions are then defined in terms of subareas. The centre subarea for each of the three hotspots is randomly determined using a uniform random distribution for both the x and y coordinates. Taxis and passengers are then randomly allocated to each hotspot using a uniform distribution and the subarea position for each taxi and passenger is then determined by using Poisson distributions with $\lambda = 70, 35, 35$, respectively, to determine the x and y subarea offset from the hotspot it is assigned to.

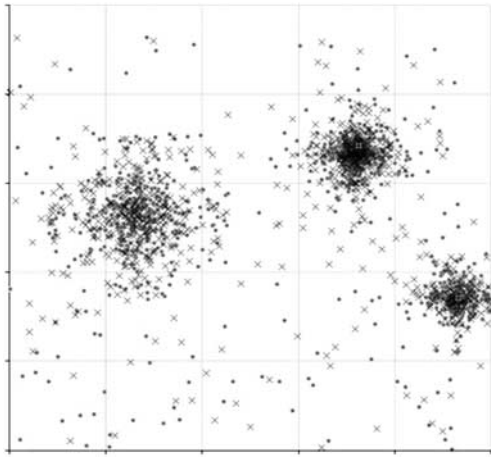


Figure 4 Distribution of 1000 taxis and 1000 passengers in a square area (taxis are denoted by *x* marks and passengers are denoted by dots).

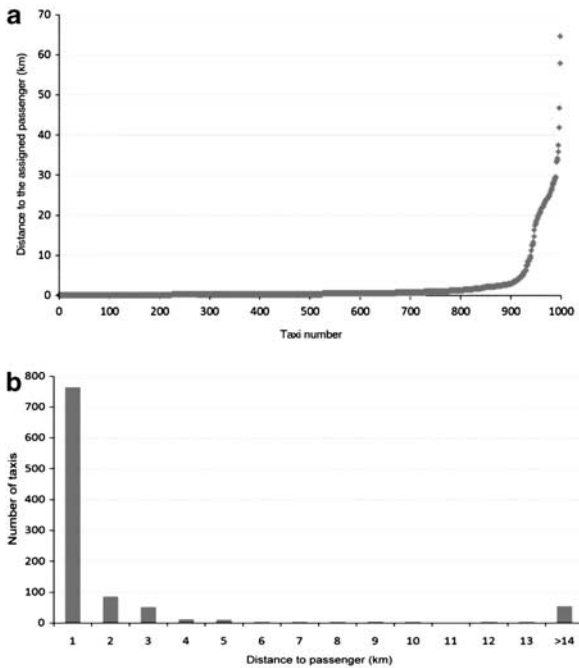


Figure 5 Optimal stable matching of 1000 taxis and 1000 passengers. (a) Distances to passenger of 1000 taxis. (b) Statistic of number of taxis corresponding to travel distance.

The optimal stable matching for 1000 pairs of taxi and passenger was computed by a brute-force search in our implementation and the distances between the taxis and the passengers are shown in Figure 5(a), ordered from smallest to largest. The smallest distance is zero and the largest distance is 64.9 km. Figure 5(b) shows the number of taxis that fall into different distance ranges. It shows that most of the scheduled taxis travel only short distances to their passengers, while a small number of taxis need to travel long distances (more

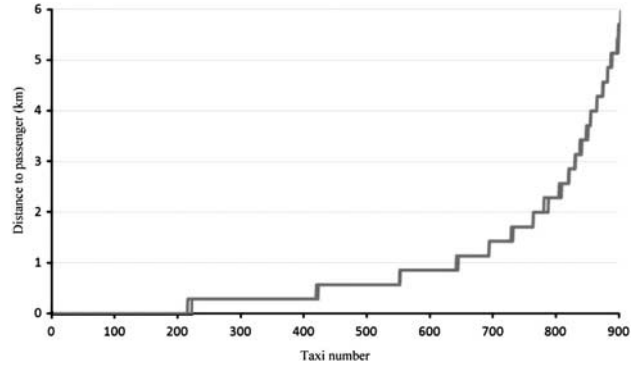


Figure 6 A comparison between the optimal stable matching (red line) and a non-optimal stable matching (blue line).

than 14 km). The computational time required for the best stable matching depends upon how many equilibria (stable matchings) a scheduling problem has. In the best case, the time complexity is $O(mn \times \log(mn))$, if there is only one stable matching. In the worst case, the time complexity may be up to $O(n^m)$. In case of the problem in Figure 5, the computational time for a non-optimal stable matching solution was 0.4 s on a PC with a dual 2.66 GHz Intel CPU, while the computational time to achieve the optimal stable solution using the brute-force search was 24.5 min.

We note that this scheduling system only provides decision support for taxi drivers. Taxi drivers will not necessarily choose to collect an assigned passenger or use the assigned route. In practice, a taxi should not be assigned a passenger who is very far away because doing so leads to high, non-profitable mileage for the taxi driver and a possibly unacceptable waiting time for the passenger. Experiments are executed to compare the optimal stable solutions and the stable equilibria obtained by the previously described algorithm. We observed that, if we ignore the taxis that are assigned to distant passengers, and left them to be scheduled in the next time window, there was little difference between the optimal stable matching and the stable equilibrium obtained by our algorithm. In Figure 6, the taxis are sequenced according to their distances to assigned passengers. The red line indicates the optimal stable matching and the blue line a non-optimal stable matching by our algorithm. It shows that the difference between the two matchings is minimal if we only consider 90% of the taxis that are assigned to passengers with distances ≤ 6 km. The reason is that most taxis and passengers are located around the hot spots and the resulting short distances will be prioritized by the pairing method used to achieve a stable matching. The amount of computation could be greatly reduced if we do not need to find the optimal stable matching and these results imply that this is practical.

In order to investigate the influence of the number of taxis and the settings of the Poisson coefficients on the gap between the optimal stable matching and non-optimal stable matchings, we performed a series of simulations on a simulated 100 km \times 100 km square area. The number of taxis is set to be

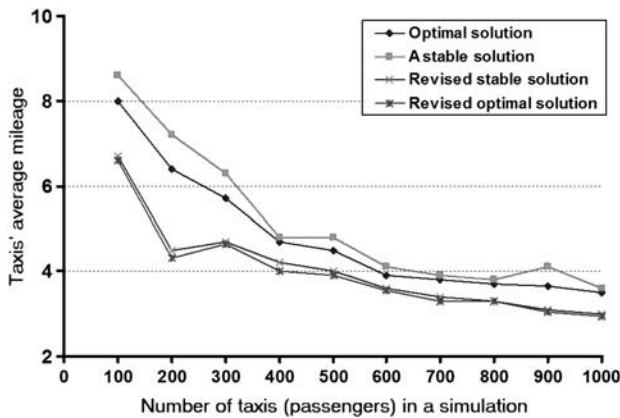


Figure 7 The effect of the number of taxis on the taxis' average mileage.

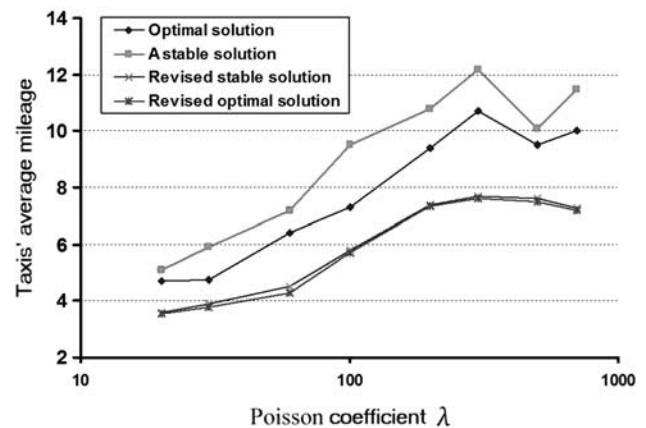


Figure 8 The influence of the Poisson coefficient on the taxis' average mileage.

equal to the number of passengers. Taxis and passengers are randomly distributed, according to a Poisson distribution. The optimal stable matching with different numbers of taxis (from 100 to 1000) and different Poisson parameters λ (from 30 to 700) is computed. Figure 7 compares the average mileage of taxis on the optimal stable solution and a non-optimal stable solution, given a Poisson distribution with $\lambda = 70$. Results for a revised optimal stable solution and a revised stable solution, which discards any taxis with an assigned mileage ≥ 50 km, are also compared. We note that taxi drivers are not forced to pick up passengers, and the drivers become increasingly unlikely to do so as the mileage increases. Figure 8 shows the influence of different Poisson parameters for a fixed number of taxis (200). The average mileage of matching is low when the distribution of taxis and passengers is centralized (the λ value is low). On the other hand, the average mileage is high when taxis and passengers are scattered. The difference between the revised solutions can be observed to be very limited in both figures.

Although taxis that are assigned to distant passengers would not necessarily pick up the passengers, the system could still be of use to these taxi drivers because it provides information about the distribution of passengers and other taxis. It can, therefore, help the taxi driver to choose where to go to next to find the next passenger.

For comparison, a first come first serve (FCFS) strategy was developed where the passengers are allocated to taxis according to the time sequence of their requests, and each passenger is assigned to the nearest taxi that was available. FCFS is a 'fair' algorithm for passengers but it increases the average vacant mileage for taxi drivers compared with the optimal stable matching where taxi drivers cooperate to minimize the average vacant mileage. We show the comparison of the FCFS strategy against the optimal stable matching in Figure 9.

4.2. Algorithms for dynamic scheduling

In practice, taxi scheduling is a dynamic process and the time window for the request of a taxi service needs to be taken

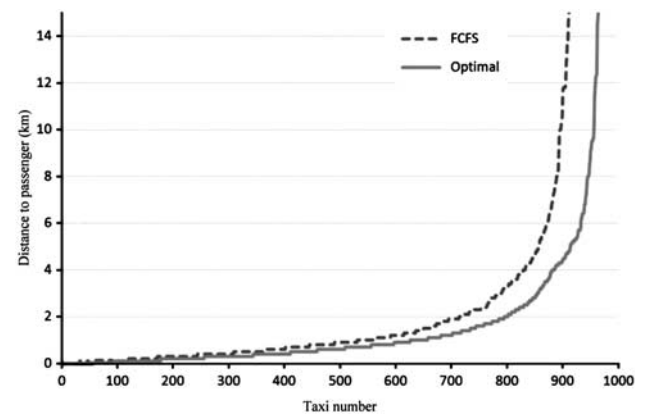


Figure 9 Comparison of the optimal stable matching with the result of FCFS strategy (the dashed line denotes the result of the FCFS strategy).

into consideration. We simulated a scheduling problem of 5000 taxis and 50 000 passengers and present the results here. The location of taxis, passengers, and their destinations were within a square area. Travel times were calculated based upon the distance between the taxi and customer positions. Of course, taxi drivers are able to reject passengers (for example if a passenger was too far away) and would normally do so straight away in that case. The travel times for taxis (and the waiting times for serviced passengers) would then be lower but there would be un-serviced passengers. We assume for these experiments that taxi drivers do not reject passengers and therefore some passengers have relatively large waiting times. In these experiments, the requests for a taxi service from the passengers were randomly distributed in a period of 240 min, in addition to the random distributions of their locations and destinations. The time of each request was determined using a Poisson distribution

with $\lambda = 150$. Three strategies are studied, which are as follows:

1. FCFS. The nearest vacant taxi will be assigned immediately when the request for service is received.
2. Strategy 1: The scheduling is performed every t_w minutes, where t_w is a parameter specifying a service window size. For these experiments, t_w was set to 5 min. The passengers whose requests for service are received within the 5 min window are all scheduled together using the algorithm described in Section 3. In practice, the value of t_w can be adjusted depending on how *busy* the system is.
3. Strategy 2: A combination of the FCFS strategy and strategy 1. A passenger will be immediately served according to the FCFS strategy if there is a vacant taxi within $l_w = 10$ km distance (l_w being another adjustable algorithm parameter), otherwise, strategy 1 will be adopted and a taxi will be allocated at the end of the current service window.

The algorithm was coded in Visual C++ and experiments were run on a PC with dual 2.66 GHz Intel CPU and 3.25 GB RAM. The results of this simulation are shown in Table 1. With strategy 1, there is a delay between the passenger requesting service and the service being scheduled (of up to 5 min in the worst case). Although the average vacant mileage is decreased for the taxis compared with FCFS, the saving in passenger waiting time is counteracted by the scheduling delay. Strategy 2 is better than the other strategies in terms of its low waiting time of passengers.

The waiting times of passengers, mileages of taxis, and vacant mileages of taxis are shown in Figures 10–12, respectively.

The results of the simulation show that the proposed scheduling algorithm benefits both taxi drivers and passengers. It increases valid mileages and decreases vacant mileages for taxis. Low vacant mileages lead to low waiting times for passengers.

We note that the settings for the simulation, the Poisson distribution of taxis/passengers and the time window t_w , have a significant influence on the results of the simulation. If the distribution of passengers is scattered, the vacant mileage of taxis will be high regardless of the scheduling strategy.

Table 1 Result of simulation

Strategy	Taxis		Passengers
	Average mileage	Average vacant mileage*	Average waiting time†
FCFS	48.28	9.70	18.5
Strategy 1	52.01	6.45	19.1
Strategy 2	51.79	6.75	14.3

*Vacant mileage: travel distance for a taxi to reach the location of passengers.

†Waiting time: the time interval between receiving of the request of service and arrival of taxi.

The smaller the value of t_w , the faster the service request will be served. In the extreme case, strategies 1 and 2 become FCFS when $t_w \rightarrow 0$. However, the quality of solutions would then be low. On the other hand, taxis are more likely to be assigned to nearer passengers when the value of t_w is higher, but the waiting time of passengers will be longer in this case. Strategy 2 is actually a tradeoff between FCFS and strategy 1. Strategy 2 will be FCFS if $l_w \rightarrow \infty$ and it will be strategy 1 if $l_w \rightarrow 0$.

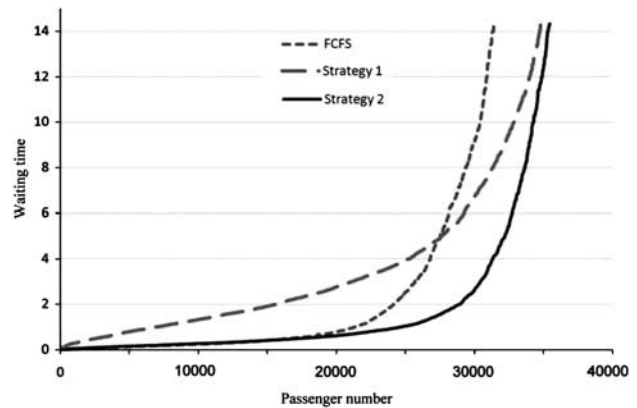


Figure 10 Waiting times for passengers, (the passengers with waiting time more than 14 min are not shown).

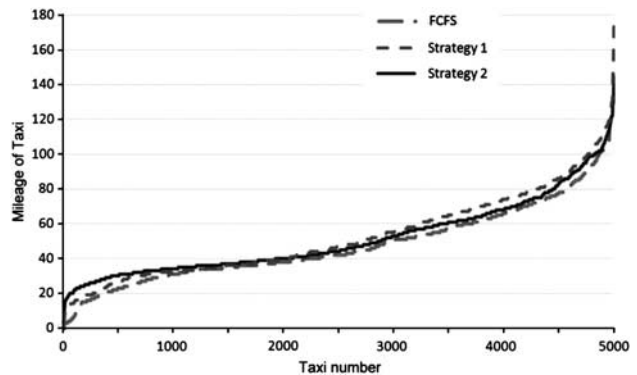


Figure 11 Mileages for the 5000 taxis.

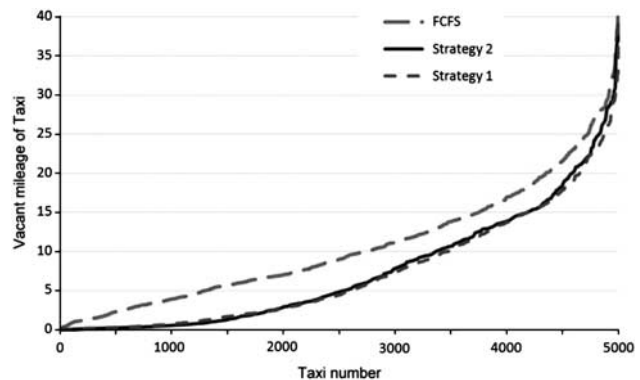


Figure 12 Vacant mileages for the 5000 taxis.

5. Practical implementation issues

The implementation of a live system to allocate taxis to passengers has to handle a number of practical issues that have not yet been discussed. First, does the algorithm actually produce a stable matching in the case of equal taxi times? Second, can travel times be calculated accurately enough for use in the allocation algorithm? Third, if taxis have identical travel times to a passenger then how does the algorithm determine which taxi to allocate?

For the first issue, consider the case in Figure 13, where taxis T_1 and T_2 are equidistant from passenger P_1 , and, without loss of generality, assume that taxi T_1 is allocated to passenger P_1 and taxi T_2 is allocated to passenger P_2 . If taxi T_2 has full awareness of the situation (that passenger P_1 exists, is closer to taxi T_2 than is passenger P_2 and that taxi T_2 is as close to passenger P_1 as taxi T_1 is, so that both taxis have an equal chance of getting to P_1 first) and passengers have no taxi preference other than taking the first taxi to appear, then taxi T_2 could, in theory, race taxi T_1 for passenger P_1 , and could, in some cases, increase its average payoff by doing so. A live system would have to resolve this issue, where taxis have similar travel times. Obvious solutions are to control information (so that a taxi driver does not have sufficient information to work out that there is a potential benefit from enforcing an assignment, or at least to ensure that the allocated taxi driver has the information first, to allow them to get to the passenger first) or to penalize in some way a taxi that races for a passenger (for example to lower the preference for that taxi in future).

The answer to the second question, of whether taxi travel times can be calculated sufficiently accurately, is partly a question of the accuracy of the model that is used, but the question actually has significant similarities to the first question. As long as there is some way to give a preference to the allocated taxi, then some inaccuracies in travel times can be ignored as far as the stability of the schedule is concerned, since the allocated taxi would get there first even if the predicted travel time was not accurate.

The third question, of how to allocate taxis to passengers, when the distances are identical, could be handled in a number of ways (here we assume that travel time is proportional to distance). First, a priority index could be introduced for each taxi and passenger, which could be updated based upon their ‘discipline’ records, so that more reliable taxis or passengers

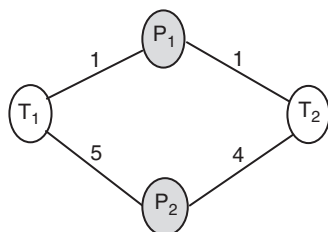


Figure 13 Example of a problem where two taxis have identical times to the passenger.

were given preference. This could also be implemented to penalize taxis that ‘race’ for passengers who have been allocated to other taxis, as was discussed in the first question. A second method could consider the ‘search tree’ of possible allocations of taxis to passengers, ‘looking ahead’ to see which allocation would be better overall. For example to consider the question, ‘If taxi T_1 was allocated, how much would taxi T_2 be penalized in terms of the extra distance to its allocated passenger, and vice versa for the penalty that taxi T_2 would have?’. This would potentially ensure that the system is fair for the other taxi drivers.

6. Conclusion and future research

Inspired by street hailing taxi services, we have proposed a novel, low cost system for the efficient allocation of independent taxis to passengers based on a game-theoretic model. Relying on mobile phones with GPS functionality, the system collects information about the locations of passengers and the locations and statuses of taxis and provides decision support for both taxi drivers and passengers. The simulation results in this paper show that our system could greatly improve the efficiency of street hailing.

The scheduling algorithm is based on a game theoretic model that considers the problem as a non-cooperative game between taxi drivers. An algorithm has been proposed, which will discover a Nash equilibrium for this game, such that each taxi driver cannot find a better choice than their assigned passenger and route. The results of our simulation show that this algorithm is efficient in computing the stable matching of taxis and passengers and it only needs small amount of computations. Thus, it is suitable for online taxi scheduling.

In addition to the time to pick up passengers, other factors may also influence the taxi drivers, for example, local knowledge of traffic conditions or preference for a specific area. Profitability and spatial equilibrium distribution of vacant taxis are discussed in (Yang *et al.*, 2010a), where some taxi drivers have private information for their decision-making and this private information is not taken into consideration in the proposed scheduling system. In these cases the effectiveness of the system will be negatively influenced. It is worth taking these subjective factors into consideration, where possible, in implementing a practical system, although doing so will increase the complexity of the algorithm. Furthermore, there is always uncertainty in real-world taxi scheduling. A passenger may change their plans or lose patience while waiting. Or a taxi may be late because of unexpected traffic congestion. The uncertainty on the proposed scheduling algorithm, and its robustness in a real-world environment in the presence of uncertainty, will form part of our future research into this interesting and important problem.

Acknowledgements—This work is supported by National Natural Science Foundation of China (Grant No. 71001055), Zhejiang Provincial Natural Science Foundation (Grant No. Y1100132) and ‘Qian Jiang’ Talent Initiative Program (QJD0802001).

References

- Bailey W and Clark T (1987). A simulation analysis of demand and fleet size effects on taxicab service rates. *Proceedings of the 19th Conference on Winter Simulation (WSC'87)*, ACM: New York, pp 838–844.
- Cairns R and Liston-Heyes C (1996). Competition and regulation in the taxi industry. *Journal of Public Economics* **59**(1): 1–15.
- Corbo J and Parkes D (2005). The price of selfish behavior in bilateral network formation. In: *ACM Symposium on Principles of Distributed Computing*, ACM: New York, pp 99–107.
- Cordeau J and Laporte G (2007). The dial-a-ride problem: Models and algorithms. *Annals of Operations Research* **153**(1): 29–46.
- Deng C, Ong H, Ang B and Goh T (1992). A modelling study of a taxi service operation. *International Journal of Operations & Production Management* **12**(11): 65–78.
- Flores-Guri D (2003). An economic analysis of regulated taxicab markets. *Review of Industrial Organization* **23**(3–4): 255–266.
- Flores-Guri D (2005). Local exclusive cruising regulation and efficiency in taxicab markets. *Journal of Transport Economics and Policy* **39**(2): 155–166.
- Gale D and Shapley LS (1962). College admissions and the stability of marriage. *American Mathematical Monthly* **69**(1): 9–14.
- Glaschenko A, Ivaschenko A, Rzevski G and Skobelev P (2009). Multi-agent real time scheduling system for taxi companies. *Proc. Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Budapest, Hungary, pp 29–36.
- Gusfield D and Irving R (1989). *The Stable Marriage Problem: Structure and Algorithms*, Vol. 54. MIT Press: Cambridge, MA.
- Halldorsson M *et al* (2003). Approximability results for stable marriage problems with ties. *Theoretical Computer Science* **306**(1–3): 431–447.
- Irani S, Lu X and Regan A (2004). On-line algorithms for the dynamic traveling repair problem. *Journal of Scheduling* **7**(3): 243–258.
- Ishii M, Lee P, Tezuka K and Chang Y (2013). A game theoretical analysis of port competition. *Transportation Research Part E* **49**(1): 92–106.
- Kendall G and Li J (2013). Competitive traveling salesmen problem: A hyper-heuristic approach. *Journal of Operational Research Society* **64**(2): 208–216.
- Lee D, Wang H and Cheu R (2003). Trip-chaining for taxi advance bookings: A strategy to reduce cost of taxi operation. *Proceedings of the 83rd Annual Meeting of the Transportation Research Board*, Washington DC.
- Li S (2006). *Multi-attribute taxi logistics optimization*. MSc Thesis, Massachusetts Institute of Technology.
- Liao Z (2003). Real-time taxi dispatching using global positioning systems. *Communication of the ACM* **46**(5): 81–83.
- Luo Y and Schonfeld P (2007). A rejected-reinsertion heuristic for the static dial-a-ride problem. *Transportation Research Part B: Methodological* **41**(7): 736–755.
- Madsen O, Ravn H and Rygaard J (1995). A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of Operations Research* **60**(1): 193–208.
- Nash J (1951). Non-cooperative games. *The Annals of Mathematics. Second Series* **54**(2): 286–295.
- National Bureau of Statistics of China (2006). Yearbook of China, chapter 16: Transportation & Communications.
- Nkoane S (2009). *Operations research modeling of the taxi transport between Polokwane and Mankweng in the Limpopo Province South Africa*. MSc Thesis, University of Limpopo.
- Prakash S and Kulkarni M (2003). Fleet management: A GPS-GIS integrated approach. In *Proceedings of the Map India Conference*, GIS Development: India.
- Yang H and Yang T (2011). Equilibrium properties of taxi markets with search frictions. *Transportation Research Part B: Methodological* **45**(4): 696–713.
- Yang H, Fung C, Wong K and Wong S (2010a). Nonlinear pricing of taxi services. *Transportation Research Part A: Policy and Practice* **44**(5): 337–348.
- Yang H, Leung C, Wong S and Bell M (2010b). Equilibria of bilateral taxi-customer searching and meeting on networks. *Transportation Research Part B: Methodological* **44**(8): 1067–1083.

Received 2 January 2013;
accepted 10 July 2013 after two revisions