Intl. Trans. in Op. Res. 13 (2006) 195–208

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Solution of the Dial-a-Ride Problem with multi-dimensional capacity constraints

K. I. Wong[a] and Michael G. H. Bell[b]

[a]*Department of Transportation Technology and Management, National Chiao Tung University, Hsinchu, Taiwan,*
[b]*Department of Civil and Environmental Engineering, Imperial College London, London SW7 2BU, UK*
E-mails: m.g.h.bell@imperial.ac.uk [Bell]; kiwong@mail.nctu.edu.tw [Wong]

Received 1 October 2005; received in revised form 13 February 2006; accepted 1 March 2006

Abstract

The Dial-a-Ride Problem (DARP) consists of planning routes and schedules for picking up and delivering users within user-specified time windows. Vehicles of a given fleet with limited capacity depart from and end at a common depot. The travel time of passengers cannot exceed a given multiple of the minimum ride time. Other constraints include vehicle capacity and vehicle route duration. In practice, scheduling is made more complicated by special user requirements and an inhomogeneous vehicle fleet. The transportation of elderly and handicapped people is an important example, as space for wheelchairs is limited and a lift is required. In this study, we present a modified insertion heuristic to solve the DARP with multi-dimensional capacity constraints, and the performance of the proposed algorithm is tested in simulation. We show that the proposed methodology is effective when compared with the classic algorithms.

Keyword: Dial-a-Ride; capacity constraints; elderly transportation

1. Introduction

The Dial-a-Ride Problem (DARP), which is also referred to as a Vehicle Routing Problem with Pickup and Delivery and Time Windows, consists of planning routes and schedules for picking up and delivering users within pre-specified time windows. Vehicles of a given fleet with limited capacity depart from and end at a common depot. The travel time of passengers cannot exceed a given multiple of the minimum ride time. Other constraints include vehicle capacity and vehicle route duration. The objective of the operator is to minimize the total service cost, minimize the inconvenience of passengers, maximize the number of users, or a combination of these. DARP is also categorized as a Pick-up and Delivery Problem with Time Windows (PDPTW), with particular relevance to transportation by trucks and courier services. In contrast, the DARP is attributed to the operational constraints for people transportation, in which adequate service

Published by Blackwell Publishing, 9600 Garsington Road, Oxford, OX4 2DQ, UK and 350 Main St, Malden, MA 02148, USA.

should be provided. In practice, scheduling is made more complicated by special user requirements and an inhomogeneous vehicle fleet.

The transportation of elderly and handicapped people is an important example. Wheelchair users may need to be transported by vehicles equipped with special facilities, e.g., a lift, for those who have to remain seated in their wheelchairs. Minibuses with accompanying personnel are preferred by those who have difficulty getting on and off the vehicle by themselves. People going for shopping may require extra space for luggage on their return trip. While there are a substantial number of studies for the multi-vehicle DARP problem, little has been said about the multiple dimensional nature of vehicle capacity. In this study, we present a modified insertion heuristic to solve the DARP with multi-dimensional capacity constraints, and then a local post-optimization technique is implemented to further reduce the objective function. The performance of the proposed algorithm is tested in simulation, and we show that the proposed methodology is more effective than classic algorithms.

2. Literature review

There is plentiful literature on the DARP and the Handicapped Transportation Problem (HTP), which is characterized by narrow time windows and vehicle capacities. The previously proposed solution methods for the DARP and HTP can be broadly classified into exact methods and heuristic algorithms. Exact methods work on the mathematical formulation of the problem and aim to find the optimal solution to the problem with optimization algorithms. However, the NP-hardness of the problem may render a complete solution of large problems unattainable within a reasonable time. On the other hand, heuristic algorithms seek a ‘‘good’’ solution within an acceptably short time. Heuristics are resorted to by the majority of studies.

For the exact methods, Psaraftis (1980) formulated the single-vehicle DARP as a dynamic programming problem with the objective of minimizing the route operating time and passenger dissatisfaction. Time windows are not explicitly considered but instead a ‘‘maximum position shift’’ constraint is imposed in the model, limiting the difference between the position of a user in the calling list and its position in the actual route. Psaraftis (1983) later extended the approach for user-specified pick-up and delivery time windows. Sexton and Bodin (1985a, b) studied the single-vehicle problem with the objective of minimizing customer inconvenience. Benders’ decomposition procedure was applied to solve the formulation resulting in a routing and scheduling heuristic.

The solution methods for the single-vehicle DARP have been sought as a step in going to the multiple-vehicle case. One of the first and frequently cited heuristics for the multiple-vehicle version of the DARP was Jaw et al. (1986). In their model, customers booking in advance can specify either a desired pick-up time or desired delivery time. Their model allows the actual pick up or delivery of a customer to deviate from the desired one, but under the constraints of a fixed maximum-wait time window and a maximum ride time that a passenger may spend in the vehicle. For the purpose of passenger transportation, vehicles are not allowed to idle when carrying passengers. The insertions of customers into the work schedule are selected while minimizing the objective function, which is a weighted sum of disutility to the customers and to the operator. They also suggested an alternative insertion procedure to improve the performance of the algorithm by making it less "myopic". Instead of processing one request at a time, a user-specified number of requests can be inserted into one assignment, although they concluded that this did not result in significant improvements but led to considerable increases in computing time.

Madsen et al. (1995) presented a heuristic algorithm with multiple capacities and multiple objectives as well as a dynamic updating capability, with an application to a real case of scheduling for elderly and disabled persons. Routes are pre-planned for the requests known at the beginning of the day, and new requests can enter the system throughout the day. Also, travel time updates and vehicle breakdowns can be considered. Inspired by the methodology of Jaw et al. (1986), the algorithm they developed can be efficient enough to be implemented in a dynamic environment for online scheduling.

Ioachim et al. (1995) showed successful results by using mathematical programming techniques to generate a set of mini-clusters in solving the multi-vehicle PDPTW. Toth and Vigo (1997) presented another piece of work on handicapped person transportation, where additional operational constraints are considered. Passengers specify the time instant for pick-up or delivery and the total travel time must not exceed a maximum travel time. Additional service times are also included at the origin and the destination, and the service requirements of each user are categorized by their physical characteristics, e.g., walking or wheelchair users. The mixed fleet consists of minibuses and special cars with capacities, while unrouted passengers can be served with taxis subject to a penalty of higher cost. A parallel insertion heuristic algorithm is used to solve the problem and a tabu thresholding optimization procedure is implemented to improve the solution after the insertion steps.

In a recent study, Diana and Dessouky (2004) presented a new heuristic for the static version of the DARP with time windows. They developed a route initialization procedure by inserting an initial request to each of the vehicles, taking the spatial and temporal effects into account. A parallel regret insertion heuristic is then used for the rest of the requests not inserted into the initialization. Instead of ranking the requests by certain criteria (e.g., earliest pick-up time or latest delivery time) as in classic insertion heuristics, the regret insertion builds up an incremental cost matrix for each of the unassigned requests when assigned to each of the existing vehicle routes. A regret cost, which is a measure of the potential difficulty if a request is not immediately assigned, is calculated for each request, and the algorithm seeks the request with the largest regret cost, and inserts it into the existing schedules. The whole procedure is repeated until all requests are inserted. In their study, vehicle fleets are assumed to be without capacity constraints. They have successfully implemented the algorithm for a real case of up to 1000 requests, resulting in an 8% cut in the fleet compared with the classic sequential insertion of Solomon (1987). However, for 100 requests they concluded that the improvement in the solution quality using the regret insertion is marginal.

Metaheuristic algorithms have been sought to solve the DARP. Cordeau and Laporte (2003a) formulated and solved the static case with a tabu search heuristic. Instead of measuring disutility by the deviation between the actual pick-up/drop-off times and the user-desired ones, their model allows users to specify a time window of a fixed width on their inbound or outbound trips, with an upper limit on the travel time for any user. Their work has later been extended to the dynamic case by Attanasio et al. (2004), where requests are received throughout the day, and solved by a parallel tabu heuristic. Mitrovic-Minic et al. (2004) considered the dynamic problem with a double-horizon-based heuristic, considering a short-term and long-term horizon. While short-term decisions are taken with the insertion procedure, improvement is obtained through a longer-term consideration performed with a local search heuristic (tabu search), efficiently using the slack period between the horizons. The methodology is constructed for the PDPTW, which is more likely faced by freight distributors and couriers, in which demands span a long period with a high proportion of immediate requests. An excellent and comprehensive review of the features and variants of the DARP is presented by Cordeau and Laporte (2003b). They surveyed over 30 publications and summarized the important algorithms that have been published over the last 30 years.

3. The DARP problem

We consider a static case in this paper. In the following, we present the problem scenario described by Jaw et al. (1986) and Diana and Dessouky (2004). A dial-a-ride system assumes that an operator managing a fleet of vehicles is responsible for providing transportation services to a group of customers. A table of trip characteristics is listed in Table 1. Each request i is associated with an origin O_i where the passenger would be picked up and a destination D_i where the passenger is to be delivered. They are also characterized by a user type r_i, specifying the type of vehicles they need. The passengers specify either a desired pick-up time (DPT_i) or a desired delivery time (DDT_i), and the scheduler attempts to meet the passenger's desired time within a maximum wait time window (WS), as described in Fig. 1. At each stop of the vehicle, additional service times for loading or unloading passenger at their origin and destination, p_i and q_i, will also be taken into account. The total time that the passenger spends on the vehicle is subject to a maximum ride time (MRT_i) constraint, which is proportional to the direct ride time (DRT_i), the travel time from the origin to destination directly without deviation. We adopted the definition of MRT_i described by Diana and Dessouky (2004) as

$$MRT_i = \max(a \cdot DRT_i + b, DRT_i + WS), \tag{1}$$

where a and b are user-specified parameters (e.g., $a = 1.5$ and $b = 5\,\text{min}$).

Table 1
Trip characteristics

O_i	Origin location of request i
D_i	Destination location of request i
DPT_i (DDT_i)	Desired pickup (or delivery) time of request i
WS	Maximum time window at pickup or delivery
DRT_i	Direct ride time of request i
MRT_i	Maximum ride time of request i
EPT_i	Earliest pick-up time of request i
LPT_i	Latest pick-up time of request i
EDT_i	Earliest delivery time of request i
LDT_i	Latest delivery time of request i
p_i	Additional service time for loading at origin of request i
q_i	Additional service time for unloading at destination of request i
r_i	User-type of request i (e.g. 0 = walking, 1 = wheelchair)

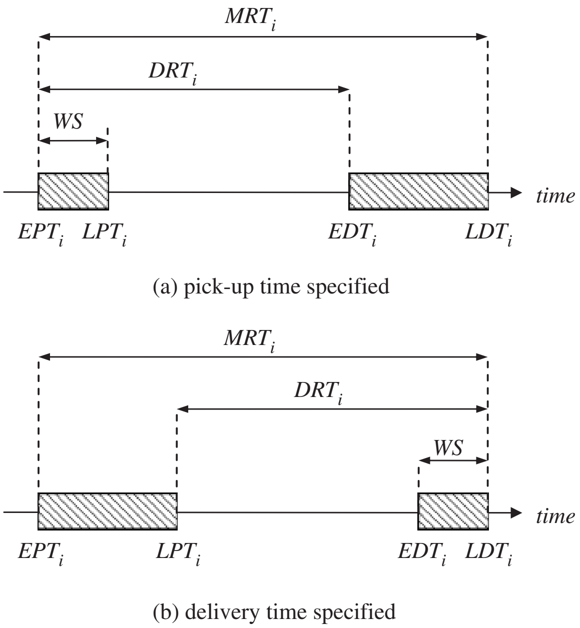

Fig. 1. Time window specifications for requests with specified (a) pick-up time or (b) delivery time.

The time windows associated with each request are related to the service quality and should be satisfied in the schedule by the operator. Let EPT_i and LPT_i be the earliest time and latest time that a request i can be picked up, and EDT_i and LDT_i be the earliest time and latest time that the request i must be delivered. A round trip is a pair of trips consisting of two single trips: an outward trip and a return trip. For the outward trip, the passenger is departing from home heading to the destination, and for the return trip he is returning home. The return trip specifies the DPT_i, and the associated time windows are determined as

$$EPT_i = DPT_i, \tag{2}$$

$$LPT_i = EPT_i + WS, \tag{3}$$

$$EDT_i = EPT_i + DRT_i, \tag{4}$$

$$LDT_i = EPT_i + MRT_i, \tag{5}$$

For the outward trip, the DDT_i is specified and the associated time windows are calculated as

$$LDT_i = DDT_i, \tag{6}$$

$$EPT_i = LDT_i - MRT_i, \tag{7}$$

$$LPT_i = LDT_i - DRT_i, \tag{8}$$

$$EDT_i = LDT_i - WS. \tag{9}$$

The intervals of pick up and delivery are illustrated in Figs 1(a) and 1(b), and any insertion that satisfies the constraints is considered to be feasible. The DPT_i-specified time windows and the DDT_i-specified time windows have different characteristics in the overall scheduling, because of the differences in the widths of the time windows at the pick-up location and delivery location. In the pick-up time-specified request, the wait time width at the origin is a constant WS, while the wait time width at the destination equals $MRT_i - DRT_i$, which relates to the travel time and therefore the traffic condition, considering the fact that there is a higher variance in travel time for a longer trip in reality. For the delivery time-specified request, the wait time width at the origin is a function of the direct travel time while the time width at the destination is a constant.

It is also noted that the computation of Equations (2)–(9) overspecifies the time constraints and provides a narrower time window compared with the problem requirement (see Diana and Dessouky, 2004). In contrast, Jaw et al. (1986) adopted a different definition but wider time windows that were used as a set of necessary but not sufficient conditions for the feasibility check of each insertion, and therefore, the delivery time window depends on the actual time the passenger is scheduled to be picked up. This time-window definition probably gives a higher quality solution to the operator. On the other hand, the definition adopted in this paper narrows the time windows down to a set of ''fixed'' values. We argue that this ''fixed'' specification of time windows allows the passenger to estimate their actual arrival time better. Recent studies using mathematical programming approaches (Fabri and Recht, 2005) or tabu search approaches (Cordeau and Laporte, 2003a) use this type of fixed time windows.

The operator schedules a mixed fleet of k vehicles, made up of a limited number of capacitated cars and minibuses equipped for wheelchair access. The vehicles are associated with a common or different depots. There are two types of passengers: wheelchair users and non-wheelchair users. For each vehicle, we consider two types of capacity: a capacity for seated persons C_k^1 and a capacity for open wheelchairs C_k^2. These two capacities are assumed to be non-substitutable throughout this study. Nevertheless, this assumption can be relaxed to allow one-way substitution (where a folded seat can be opened in a wheelchair space) or two-way substitution. Each of the vehicles is also associated with a fixed cost Y_k for each initialization and a routing cost K_k for each unit distance or time the vehicle is utilized. A list of vehicle characteristics is presented in Table 2. Each customer is attributed with a service requirement, depending on his/her physical condition, and therefore can only be delivered with vehicles with certain equipment. For instance, Toth and Vigo (1997) categorized users into three groups for handicapped people transportation, as listed in Table 3. Wheelchair users who cannot walk and have to remain seated can only be transported by vehicles equipped with an elevator, while those who can walk and sit in a normal seat can be served with any vehicle but may need a space to park the closed wheelchair. Passengers who require assistance can only be assigned to vehicles with specialized crew.

Furthermore, the dimension of the above-mentioned constraints can vary and extend in many ways with practical considerations, adding further difficulty to the scheduling. On the passenger side, it is not uncommon that the dial-a-ride service is used for shopping by the user so that additional storage space may be needed for shopping or luggage. On the view of the operator, smaller vehicles can be used to supplement larger vehicles, so that a complex composition of the

Table 2
Vehicle characteristics

C_k^1	Capacity in seated persons in vehicle k
C_k^2	Capacity in open wheelchairs in vehicle k
Y_k	Fixed cost in initialization for vehicle k
K_k	Routing cost for each time unit for vehicle k

Table 3
Service rule for the Handicapped Persons Transportation

Use wheelchair?	Remain seated in wheelchair?	Vehicle type allowed	Loading/unloading time	Vehicle equipment
Yes	No	With elevator	Long	Wheelchair
Yes	Yes	Any	Long	Persons
No	Yes	Any	short	persons

vehicle fleet exists in the scheduling process. While smaller vehicles can be used for journeys involving one or two passengers requiring services for distinct areas, larger vehicles are suitable for collecting a number of passengers who are going to the same activity (e.g., social gathering, shopping mall, etc.).

In the model, it is assumed that requests that are not able to be inserted into the schedules will be delivered by taxis, subject to a higher cost that depends on the distance travelled. Each taxi can only carry a small group of passengers at a time, and each taxi trip is entirely independent. Unlimited number of taxis can be adopted, as the number of users in the dial-a-ride system is relatively small. The high cost can also be regarded as a penalty cost for each passenger rejected by the system.

The objective function of the problem is a weighted combination of the total operating time of the dial-a-ride fleet, the passenger delay (time extra to their direct travel time), and the cost for taxi trips for transporting the uninserted requests. The routing cost for each vehicle type is defined and used in the objective function evaluation.

4. The algorithm

In this section, the modified parallel insertion heuristic for the DARP with multi-dimensional capacities will be described. The construction of the heuristic is motivated by the characteristics of the operational constraints that are normally faced by the scheduler for people transportation, especially for older and disabled people who may need special care. The development of the algorithm is inspired by the features of the problem faced in practice.

The algorithm proposed aims to produce a quick and flexible insertion method. There are three phases in the solution algorithm. Trip characteristics are calculated and trips are ranked with a particular order for insertion. Then, parallel insertion is performed to insert the requests into the existing routes. Requests are iteratively inserted, and for those that cannot be inserted or for which insertion may incur a high cost, a taxi will be used. An optional local search procedure can be used to further optimize the objective function.

4.1. Ordering

It has been noted by many authors that the sequence or order of requests to be inserted into the schedule has a significant impact on the performance of insertion heuristics (see Diana and Dessouky, 2004; Jaw et al., 1986). The underlying idea of this initialization strategy is to find out and order those requests that are difficult to be scheduled if they are not inserted now but later. In a mixed routing and scheduling problem, a rank index can be defined in relation to a number of features. Spatial and temporal effects are important factors here. Requests that are far from the centre and far from the locations of other requests should be serviced with a higher priority. Temporal effects describe the time that a user desires to be serviced over the day, and in general, it is favorable to insert a request with an earlier pick-up time prior to a later one. The loading and unloading times of the customers during pick up and delivery should also be considered. There is also a factor concerning the dimensional capacity of the vehicles. As the empty spaces in vehicles vary during the day as passengers get on and off, the users whose requirement is relatively difficult to meet should be inserted with a higher priority. This can utilize the resources of the vehicles more efficiently in the sense that the transportation for users of a more common type can be easily covered with the help of other vehicles.

The ranking index of the requests is defined as follows: each of the requests is rated with a difficulty degree, δ_i, which measures the difficulty and inconvenience caused to other requests when the request i is inserted into the routes:

$$\delta_i = \beta_1(\overline{r_i}) - \beta_2(\overline{EPT_i}) + \beta_3(\overline{p_i + q_i}) + \beta_4(\overline{d_i}), \tag{10}$$

where β_1, β_2, β_3, and β_4 are positive weights, r_i is a parameter related to the user type, EPT_i is the earliest pick-up time, p_i and q_i are the loading and unloading times at origin and destination locations, respectively, and d_i is a degree of decentralization measuring the location of the request. As these variables are different in scale, conferring complexity in setting up the weighting factors, a function $(\overline{\ })$ is suggested to normalize the variables inside the parenthesis, by dividing the value of the variable of index i by the maximum value in the set of all requests, i.e. $(\overline{x_i}) = \left(\frac{x_i}{\max_i x_i}\right)$. The user type parameter r_i is defined as

$$r_i = 1 - \frac{\text{Number of vehicles allowed to pick up request } i}{\text{Total number of vehicles}}, \tag{11}$$

disregarding the vehicle categories. The degree of decentralization d_i is computed as the average travel time needed to connect trip i to each of the other trips j directly, and is defined as

$$d_i = \sum_{j \neq i} (t_{+i,+j} + t_{+i,-j} + t_{-i,+j} + t_{-i,-j}), \tag{12}$$

where $t_{+i,+j}$ is the direct travelling time from O_i to O_j and $t_{+i,-j}$ is the direct travelling time from O_i to D_j (+, arrival, −, departure). Heuristically, we set the values of the parameters as $\beta_1 = 0.4$,

$\beta_2 = 0.3$, $\beta_3 = 0.2$, and $\beta_4 = 0.1$, for a descending order in importance of the factors. Requests are then inserted in the descending order of δ_i.

4.2. Parallel insertion heuristic

The parallel insertion procedure is similar to the one proposed in Jaw et al. (1986). For each of request i in descending order of δ_i, find all feasible ways in which request i can be inserted into the existing route of vehicle k, and calculate the additional cost to the objective function after each feasible insertion. Feasible insertion means feasibly inserting the pick up and drop off of a request into an existing route that is feasible before the insertion. For each insertion, the delivery must occur after the pick up on the route. The vehicle would have to detour, and the sequence of pick up and drop off (or stops) along the route of the vehicle should be feasible such that the time constraint related to each of these stops and the capacity constraint of the vehicle with that particular passenger type are satisfied. With those feasible ways of insertion, the best position with the minimum additional cost is determined. If there is no infeasible way to insert this request, then the additional cost is set as an arbitrary large number. The above is repeated for each of the available vehicles, and the request is inserted into the vehicle and position with the minimum objective. In the case where the fleet is highly utilized and there is no feasible way to insert a request, a taxi is dispatched for the service. A taxi can also be used for the case where a request comes from a distant location or in an unsociable time (last minute of the day), and the use of such vehicles can cost less to the overall system as well as delay to this and other passengers.

It is also noted that the computation of the objective function builds upon a scheduling subroutine, which determines the actual arrival and departure time from and to the depot and stops along the route. Like scheduling a bus, these times can be shifted forward and backward and there could be many ways to do this. We choose to push the schedule in a way such that the vehicle departs a stop as early as possible while the time constraints of later requests are not violated. The heuristic of Jaw et al. (1986) decomposes each route into schedule blocks, which mean continuous intervals of service time without idling in between. They also developed a procedure for rapidly checking the feasibility of the insertion of a request into a particular scheduled block. When inserting a request into an existing route, the duration of a certain block may have to be extended and the pause between two consecutive blocks can be eliminated, causing the time of the two blocks to overlap. In this case, the two blocks can be merged into one. This merging involves extra calculation and the steps are not fully discussed in Jaw et al. (1986). In contrast, our heuristic does not consider the blocks, but checks the feasibility of the insertion of a node only on the basis of time windows constraints of each stop on the route. Pauses between stops are inserted only if the vehicle arrives before the earliest departure time from the stop. In our approach, the length of this idling is recomputed in every insertion. Similar to the classic insertion heuristics, the computational complexity of the proposed heuristic is $\mathbf{O}(n^2)$, where n is the number of requests.

4.3. Local post-optimization

The parallel insertion heuristic generates a set of feasible routes, processing the list of requests in sequence. As the optimality of such a procedure is difficult to quantify, a local post-optimization phase is developed to further improve the sequence. It is inspired by and analogous to the tabu thresholding technique used in Toth and Vigo (1997), in which three exchange heuristics have been discussed. "Trip insertion" removes a given trip from a route and then inserts it into the best position of another route. "Trip exchange" removes two given trips that belong to different routes, swaps them, and then reinserts them into the best position of the route of each other. "Trip double insertion" is to remove a given trip and insert it into another route, and then another trip from a third route is removed and inserted into the best position of the original route of the first trip. The computational complexity of an iteration of the procedure is $\mathbf{O}(n\tilde{n}^2)$ for trip insertion and $\mathbf{O}(n\tilde{n}^3)$ for trip exchange and trip double insertion, where n is the number of requests and $\tilde{n}$ is the average number of trips in a route. These heuristic rules have been tested in our simulation, and all of them can improve the objective function value of the solution. As the trip insertion only involves one trip at a time and trip exchange and trip double insertion involve two trips for each iteration, trip insertion involves less computing steps than the other two if one wants to try all combinations of exchanges. Furthermore, in the original work of Toth and Vigo (1997), to speed up the computing procedure, the exchange set of requests for trip exchange and trip double insertion were limited heuristically to a small number due to the computational complexity. Our initial numerical test (not shown in the paper) showed that the improvements in solution by trip exchange and trip double insertion over trip insertion are marginal. In the following computational analysis, we use trip insertion for each of the requests until the solution cannot be improved any further, i.e., reaching a local optimum.

5. Computational analysis

In this section, the computational experiment is described and the performance of the proposed algorithm will be examined. A set of hypothetical problems are generated with particular reference to real situations. The problem is solved, and a sensitivity analysis is conducted with respect to the model parameters.

We consider a problem with two-dimensional capacities here: a capacity for seated persons and a capacity for open wheelchair spaces. This can reflect the need relating to wheelchair users and non-wheelchair users. In setting up the fleet composition in the example, while all vehicles have passenger seats, only some of those have open spaces for wheelchairs. Generated by the Monte Carlo simulation procedure, the origin and destination of requests are randomly and independently distributed over a service area in a 20 km square Euclidean plane. The travel time between any two coordinates is taken to be equal to the direct distance between them divided by the average vehicle speed, which is assumed to be 20 km/h here. A single depot for the whole vehicle fleet is considered and located in the centre of the study area. The cost of the taxi trips is simply set as a linear function of the distance and time travelled but independent of where the taxi originated from, which is normally the case. Over a day, there are a total of 150 demands, with 50% being outward trips and 50% being return trips, and there is a 30% chance for an incoming request being a wheelchair user. We also assume that there are loading and unloading times of 10 min for all requests. The demand pattern is shown in Fig. 2, and the desired service times are clustered around the morning peak for the outward trips and around the evening peak for those return trips, over a simulation period of 12 h (vehicle running can be outside this period). We take

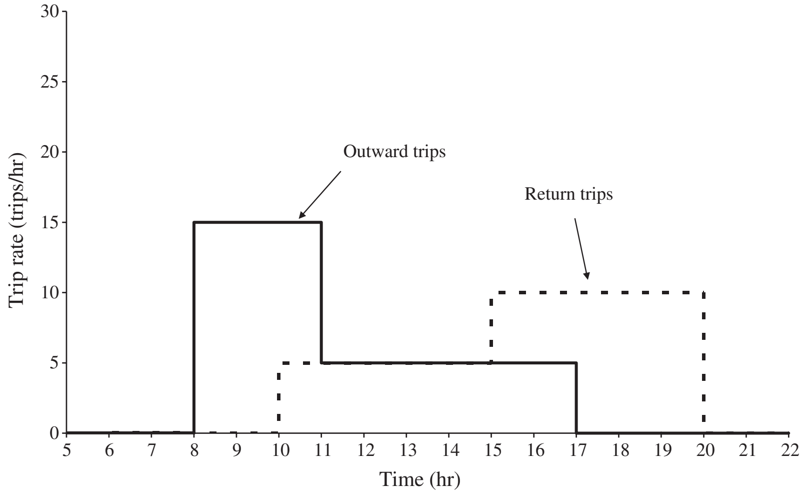

Fig. 2. Demand pattern of outward trips and return trips over the day.

Table 4
Vehicle composition in the available fleet

Type	Quantity	Capacity		Cost	
		Passenger seat	Open wheelchair	Per hour	Fixed
Minibus	7	7	3	60	0
Minibus	8	4	0	60	0
Taxi	∞	1	1	120	500

$a = 2.0$ and $b = 20$ min in the computation of maximum ride time. The vehicle composition available to the operator is given in Table 4, in which some vehicles are equipped with facilities for wheelchairs while others are not. Taxis are given high fixed and time-based charges to reduce their usage and make better use of the existing vehicles.

The problem is solved with the classic parallel insertion algorithm and the proposed algorithm, with and without a local optimization phase. The classic parallel insertion algorithm is set up with the sequence of requests being ranked by their earliest pick-up times, while the proposed algorithm is solved as described in the paper. The optional local optimization phase is performed on the output of both insertion heuristics to investigate its effectiveness. The value of the maximum wait time window *WS* is tested with 20 and 10 min in order to examine the sensitivity of system performance to this service parameter and the results are shown in Tables 5 and 6. The tables describe the number of minibuses and taxis used, the mileage and hours associated with the fleet of minibuses, average travel time and delay for passengers, the value of the objective function, and the relative improvement in the objective as compared with the basic case. From Table 5 with a *WS* of 20 min, it is shown that the proposed algorithms performed better than the classic parallel insertion. The proposed parallel insertion produces a solution with 6.4% less

Table 5
Computational results comparing the insertion heuristics, with a maximum wait time window $WS = 20$ minutes

	Vehicle used		Vehicle miles		Vehicle hours		Passenger time on board (minutes)		Objective function
Algorithm	Minibus	Taxi	Total	Empty	Ride	Idle	Average	Delay	(improvement)
1 Parallel insertion	15	14	2000.7	627.8	8373.2	1858.9	33.5	4.0	16574.2 (0.0%)
2 Parallel insertion with local optimization	15	13	1985.1	610.3	8326.4	1849.1	33.5	4.0	15977.6 (3.6%)
3 Proposed parallel insertion	15	12	2047.5	696.6	8625.5	1902.9	33.5	4.6	15508.6 (6.4%)
4 Proposed parallel insertion with local optimization	15	10	2060.4	709.2	8680.7	1899.2	33.5	4.2	14393.8 (13.2%)

Table 6
Computational results comparing the insertion heuristics, with a maximum wait time window $WS = 10$ minutes

	Vehicle used		Vehicle miles		Vehicle hours		Passenger time on board (minutes)		Objective function
Algorithm	Minibus	Taxi	Total	Empty	Ride	Idle	Average	Delay	(improvement)
1 Parallel insertion	15	23	2011.9	714.9	8467.6	1969.9	33.5	2.6	21628.4 (0.0%)
2 Parallel insertion with local optimization	15	23	1985.4	695.8	8468.5	2050.3	33.5	2.5	21557.9 (0.3%)
3 Proposed parallel insertion	15	20	2092.5	774.0	8646.9	1813.5	33.5	3.1	20205.4 (6.6%)
3 Proposed parallel insertion with local optimization	15	20	2081.8	764.2	8614.7	1813.4	33.5	3.1	20165.7 (6.8%)

objective value against the classic parallel insertion, and the local optimization step further decreases the solution with a 13.2% improvement. These improvements are mostly due to the decrease in the usage of taxis, meaning that the existing fleet of minibuses are routed more efficiently. The results also show that the local optimization technique can further optimize the solution for both insertions in heuristics.

As compared with the case in Table 5 (for $WS = 20$ min), the scenario in Table 6 (for $WS = 10$ min) requires a higher standard of services to the passengers with narrow time constraints. As the number of minibuses is limited to 15 in both cases, it is expected that the solution in Table 6 would take more taxis with higher objective values. The objective value by parallel insertion with $WS = 20$ min is 16,574.2, using 14 taxis, while it is 21,628.4 with 23 taxis for the case of $WS = 10$ min. Aiming to provide a better level of service with a smaller maximum wait time span of 10 min, the system becomes very busy. Therefore, room for improvements to the solution by local optimization is limited and not significant compared with the case of $WS = 20$ min. From Table 6, the local optimization improves the solution by classic parallel insertion from 21,628.4 to 21,557.9, a decrease of 0.3%. For the case of proposed parallel insertion, the local optimization phase advances the objective from 20,205.4 to 20,165.7, a decrease of 0.2%. The solution of the proposed parallel insertion used 20 taxis, which is three vehicles fewer than used by the classic parallel insertion. In conclusion, the proposed insertion heuristic improves the solution for all cases, but the efficiency of the post-local optimization depends on the assigned level of constraint, which is the maximum time for a passenger wait in this example. The computing time for a run was about three minutes on a Pentium 4 computer.

6. Conclusions

In this paper, we describe the DARP that is usually faced in handicapped persons transportation. The multi-dimensional issues of vehicle types and passenger requirements are addressed, and a modified parallel insertion algorithm is presented to solve the problem. A local optimization technique is also discussed that can further improve the sequencing and scheduling. For an example with two vehicle types and different capacity dimensions, the proposed algorithm shows its advantages over the classic parallel insertion heuristic.

Acknowledgements

This research was supported by a research grant from the National Science Council of Taiwan (Project No. NSC 94-2420-H-009-004). The first author gratefully acknowledges the Croucher Foundation of Hong Kong for their financial support during his stay in Imperial College London.

References

Attanasio, A., Cordeau, J.F., Ghiani, G., Laporte, G., 2004. Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing* 30, 377–387.

Cordeau, J.-F., Laporte, G., 2003a. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research* 37B, 579–594.

Cordeau, J.-F., Laporte, G., 2003b. The Dial-a-Ride Problem: Variants, modeling issues and algorithms. *4OR-Quarterly Journal of the Belgian, French and Italian Operations Research Societies* 1, 89–101.

Diana, M., Dessouky, M.M., 2004. A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. *Transportation Research* 38B, 539–557.

Fabri, A., Recht, P., 2005. On dynamic pickup and delivery vehicle routing with several time windows and waiting times. *Transportation Research B* in press.

Ioachim, I., Desrosiers, J., Dumas, Y., Solomon, M.M., Villeneuve, D., 1995. A request clustering algorithm for door to door handicapped transportation. *Transportation Science* 29, 63–78.

Jaw, J.J., Odoni, A.R., Psaraftis, H.N., Wilson, N.H.M., 1986. A heuristic algorithm for the multi-vehicle many to many advance request dial-a-ride problem. *Transportation Research* 20B, 243–257.

Madsen, O.B.G., Raven, H.F., Rygaard, J.M., 1995. A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of Operations Research* 60, 193–208.

Mitrovic-Minic, S., Krishnamurti, R., Laporte, G., 2004. Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. *Transportation Research* 38B, 669–685.

Psaraftis, H.N., 1980. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science* 14, 130–154.

Psaraftis, H.N., 1983. An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. *Transportation Science* 17, 351–357.

Sexton, T.R., Bodin, L.D., 1985a. Optimizing single vehicle many-to-many operations with desired delivery times: 1. Scheduling. *Transportation Science* 19, 378–410.

Sexton, T.R., Bodin, L.D., 1985b. Optimizing single vehicle many-to-many operations with desired delivery times: 2. Routing. *Transportation Science* 19, 411–435.

Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time windows constraints. *Operations Research* 35, 254–265.

Toth, P., Vigo, D., 1997. Heuristic algorithms for the handicapped persons transportation problem. *Transportation Science* 31, 60–71.