Exact Algorithm for Dial-A-Ride Problems with Time-Dependent Travel Cost

Ta-Yin Hu [a], Chin-Ping Chang [b]

[a,b] *Department of Transportation and Communication Management Science, National Chen Kung University, No.1,Ta-Hsueh Road, Tainan 701, Taiwan, R.O.C.*
[a] *E-mail: tyhu@mail.ncku.edu.tw*
[b] *E-mail: pingszone@gmail.com*

Abstract: To serve elderly and disabled people, developing Demand Responsive Transit Service (DRTS) with flexible routes and changeable schedules is an important work. The dial-a-ride systems are the one of practical applications of the DRTS. Although many literatures develop an algorithm to improve the computing time to obtain acceptable solution under the minimum operational cost, the influence of the time window and the traffic condition for DARP is seldom to discuss. The aim of this paper is to explore the influence of the time window and the traffic condition and apply branch-and-price approach to design vehicle routes and schedule. The proposed algorithm is tested by a sub-network of Kaohsiung City, Taiwan. In the numerical experiments, several scenarios with the different time window are conducted and evaluate experimental results by objective value, computational time, average pickup delay time and average delivery delay time.

Keywords: Demand Responsive Transit Service, Dial-a-ride problems, Branch-and-price algorithm

1. INTRODUCTION

To meet transportation needs for the elderly and disabled people, developing Demand Responsive Transit Service (DRTS) with flexible routes and changeable schedules is an important work. The dial-a-ride systems are the one of practical applications of the DRTS. Well-designed vehicle routes and schedule for dial-a-ride systems can improve transportation efficiency and save total travel costs. In Taiwan, both Taipei and Kaohsiung Cities propose to build the Fu-Kang Bus for the elderly and disabled peoples. To use the service of the Fu-Kang Bus, the elderly and disabled peoples could reserve the transportation service by telephone, internet and fax in advance. How to design an appropriate vehicle routes and schedule to achieve the certain objective is a critical issue for dial-a-ride problems (DARP). The dial-a-ride transportation service and demand responsive transit services can be generalized as DARP. The DARP is defined as follows: a fleet of vehicles with fixed capacities visit all demands with specific pickup and delivery requests. The vehicle routes and schedule must satisfy the time window constraints. All vehicles must satisfy the vehicle capacity constraints. According to the requests of the customers, the dispatcher assigns vehicles to serve all demands and designs vehicle routes under the specific objective.

For the study of the DARP, most literatures consider the minimum travel costs to construct solution framework or formulate mathematical model and then apply appropriate algorithm or heuristics to solve based on the scale and complexity of the problem. Traditional exact approach including simplex method and dynamic programming is capable of obtaining the optimal solution for DARP. However, the computational cost will increase exponentially when the size of instances expands gradually. In recent years, the new exact algorithm, branch-and-price approach has been successfully applied to handling more requests for DARP than traditional exact algorithm. Although many literatures develop an algorithm to improve the computing time to obtain acceptable solution under the minimum operational cost, the influence of the time window and the traffic condition for DARP is seldom to discuss. The aim of this paper is to explore the influence of the time window and the traffic condition and apply branch-and-price approach to design vehicle routes and schedule. The proposed algorithm is tested by a sub-network of Kaohsiung City, Taiwan. In the numerical experiments, several scenarios with the different time window are conducted and evaluate experimental results by objective value, computational time, average pickup delay time and average delivery delay time.

This paper is organized as follows: Section 2 reviews the related studies of DARP. The mathematical model for DARP is formulated in Section 3. Section 4 presents the solution framework. In Section 5, the numerical experiments are conducted in a Kaohsiung network. Conclusions and comments of this paper are discussed in Section 6.

2. LITERATURE REVIEW

Based on the characteristics of demand nodes for DARP, Psarafits (1980) and Cordeau and Laporte (2003), (2007) considered that the DARP can be categorized into two different kinds of problems: static DARP and dynamic DARP. In the static DARP, all demands are known in advance. According to the information of the given demands, the dispatcher could design appropriate routes and schedules to serve customers under the specific objective function. In contrast with the static DARP, the dynamic DARP only knew partial information of the demands and the real-time demands might reveal during the vehicle routing. The dispatcher must redesign the vehicle routes and service sequences to fulfill the real-time demands. Cordeau and Laporte (2003), (2007) surveyed several academic studies for DARP and classified DARP according to the number of vehicles and the characteristics of the demands. In terms of the number of vehicles, the DARP can be classified into the single-vehicle DARP and multi-vehicle DARP. In terms of the characteristics of the demands, the DARP can be classified into the static DARP and the dynamic DARP.

2.1 Exact Algorithm

Psarafits (1980) applied dynamic programing to solve single-vehicle DARP with static and dynamic requests. The objective function was to minimize the weighted sum of route completion time and customer dissatisfaction which incorporated customer`s waiting time and ride time. The results of the numerical experiments revealed that the dynamic programming only can solve small instances with 9 requests. Dumas et al. (1991) utilized column generation approach to deal with multi-vehicle DARP and decomposed the problem into a master problem (set partitioning problem) and a subproblem (constrained shortest path problem). The numerical experiments showed that the proposed algorithm was capable of handling the randomly generated instances up to 55 requests. Cordeau (2006) formulated the three-index DARP formulation and built the branch-and-cut algorithm to solve DARP. The numerical experiments were able to solve the size of problems with 48 requests. Ropke et al. (2007) later applied same approach to explore DARP. In this paper, the authors constructed the two-index DARP formulation and compared the difference between the two-index DARP formulation and the three-index DARP formulation. Based on the experimental results, the proposed algorithm was able to solve the size of instances with up to 96 requests. Ropke and Cordeau (2009) devised the branch and cut and price algorithm to deal with the pickup and delivery problems under the minimum total routing costs. In the research framework, the author applied three different kinds of heuristics to solve the subproblem: construction algorithm, large neighborhood search and truncated label-setting algorithm. The proposed algorithm was tested by several sizes of instances ($30 \leq n \leq 500$). Cortes et al. (2010) applied the branch-and-cut method and considered the Benders decomposition for pickup and delivery problems. The proposed algorithm was tested by several random instances with up to 6 requests. Hu and Chang (2011) formulated the time-dependent DARP under the minimum total travel time. The mathematical formulation was solved by the mathematical programming software, CPELX. The time-dependent travel time data were generated by the traffic simulation software, DynaTAIWAN. The instances were generated randomly in a Kaohsiung network. The experimental results revealed that the range of the time window obviously affects the computational time and objective value.

2.2 Heuristic Approach

Psarafits (1983) devised an $O(N^2)$ heuristic algorithm to solve single DARP. In the research framework, the author applied the minimum spanning tree to generate the initial feasible solution and then took advantage of the node exchanges to improve current solution. The experimental results showed that the proposed algorithm was capable of coping with size of instances (n=50). Li and Lim (2001) proposed a tabu-embedded simulated annealing algorithm for the pickup and delivery problem with time windows under the minimum the number of vehicles, total travel distances, total schedule time and total waiting time. The experimental results revealed that the proposed algorithm can handle the size of instances with 100 requests. Fu (2002) took the time-varying and stochastic congestion into account to build heuristic approach for DARP. In the numerical experiments, the proposed heuristic approach tested with up to 2800 requests. Diana and Dessouky (2004) presented a new regret insertion heuristic for large-scale DARP under the minimum the total travel distance, the excess ride time for all customers and total idle time in the schedule. The numerical experiments were conducted in the Los Angeles city. Based on the numerical experiments, the results showed that the new regret insertion heuristic is capable of solving large size of instances with up to 1000 requests. Fabri and Recht (2006) built an algorithm to cope with the dynamic pickup and delivery vehicle routing problem with time windows. The computational experiments revealed that the proposed algorithm is capable of solving the large size of instances with up to 1000 requests. Wong and Bell (2006) proposed the modified insertion heuristic to solve static DARP under the minimum total service costs, inconvenience of passengers and the travel costs for taking a taxi for the unserved requests. The computational results revealed that the modified insertion heuristic outperforms the classic parallel insertion heuristic. Ropke and Pisinger (2006) built an adaptive large neighborhood search heuristic (LNS) for the pickup and delivery problem with time window. The proposed heuristic was tested by several sizes of instances: 100, 200, 400, 600, 800 and 1000 requests. Xiang et al. (2006) proposed a fast heuristic to deal with a large-scale DARP under the minimum fixed costs, travel costs, driver costs, waiting costs and service costs. The results showed that the proposed heuristic is capable of handling large size of instances with up to 1000 requests. Lois et al. (2007) developed a very large scale neighborhood heuristic algorithm to solve multi-vehicle DARP. The proposed heuristic was tested with up to 28 requests. Kim and Haghani (2011) devised three different kinds of heuristic algorithm: sequential insertion heuristic, parallel insertion heuristic and clustering first-routing second heuristic. The computational results showed that sequential insertion heuristic obtain the lower objective value with short computational time than the others heuristics.

2.3 Meta-heuristic Approach

Cordeau and Laporte (2003) utilized tabu search heuristic to cope with multi-vehicle DARP. The computational experiments indicated that the tabu search heuristic is capable of solving the randomly generated instances with 24 to 144 requests. Jørgensen et al. (2007) took advantage of the genetic algorithm to handle multi-vehicle DARP under the minimum total transportation costs and the inconvenience of customers. The proposed algorithm was tested by large sizes of instances (n=144). Cortes et al. (2009) introduced the particle swarm optimization (PSO) to deal with the dynamic pickup and delivery problem. D`Souza et al. (2012) compared four different kinds of approaches including Simulated Annealing (SA), Particle Swarm Optimization (PSO), Genetic Algorithm (GA) and Artificial Immune System (AIS) for DARP. The computational results revealed that PSO has worst quality of solutions and SA tends to search local optimal solution. AIS and GA were able to solve effectively and obtain better quality of solutions.

According to the previous studies, the exact algorithm, the heuristic-based algorithm and the meta-heuristic algorithm are capable of solving DARP. The exact algorithm can search the optimal solution but the large the size of instances might result in high computational costs. In contrast with the exact algorithm, the heuristic-based algorithm and the meta-heuristic algorithm are capable of dealing with large size of instances with acceptable computational costs. However, the heuristic-based algorithm and the meta-heuristic algorithm are difficult to obtain optimal solution and almost can only search second best solution.

3. PROBLEM STATEMENT AND FORMULATION

Consider a directed graph $G = (N, A)$, which include the set of nodes N and the set of arcs A. The set of nodes $(N = P \cup D \cup \{0\} \cup \{2n+1\})$ are divided into four different kinds of node: source node, sink node, pickup node and delivery node. The notation "$P = \{1,...,n\}$" represents the set of pickup nodes and the notation "$D = \{n+1,...,2n\}$" represents the set of delivery nodes. In this research, the depot node is decomposed into source node $\{0\}$ and sink node $\{2n+1\}$. The source node only has outgoing arcs and the sink node only has incoming arcs. Each request i corresponds to a pair of nodes $(i, n+i)$ with pickup node and delivery node. The notation q_i is defined as the load at node i. The load at node i is $q_0 = q_{2n+1} = 0$ and $q_i = -q_{n+i}, \forall i \in P$. Each node i has the time window constraint $[e_i, l_i]$ and the service time s_i, the desired pickup/delivery time (DPT/DDT), directed ride time (DRT) and the maximum ride time limitation (MRT). DRT_i is the travel cost between pickup node i and corresponding delivery node $n+i$. According to the desired service time, the customers can be classified in to DPT-specified customers and DDT-specified customers. The figure of DPT-specified customers is shown in Figure 1 (a) and the figure of DDT-specified customers is shown in Figure 1(b).

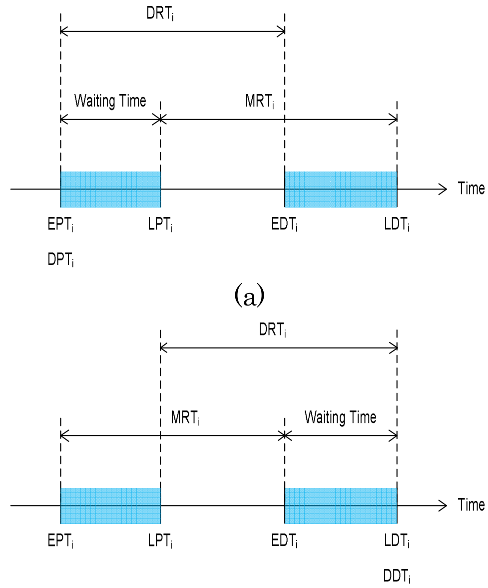

Figure 1. DPT-specified customers and DDT-specified customers (Jaw et al., 1986)

In order to consider the fluctuation of the travel time of link, this research adopts the concept of the “step function” (Malandraki and Daskin, 1992) to reflect the variation of the travel time over the time of a day. The step function decomposes the continuous travel time function into several discrete travel time functions. Each discrete travel time function corresponds to the specific time interval.

The time-dependent DARP with time windows can be formulated as the following mixed integer program:

Minimize

$$\sum_{i\in N}\sum_{j\in N}\sum_{m\in M}\sum_{v\in V} c_{i,j}^{m} x_{i,j,v}^{m} \tag{1}$$

subject to

$$\sum_{j\in N}\sum_{m\in M}\sum_{v\in V} x_{i,j,v}^{m} = 1 \qquad (i\in P), \tag{2}$$

$$\sum_{i\in P}\sum_{m\in M} x_{0,i,v}^{m} = 1 \qquad (v\in V), \tag{3}$$

$$\sum_{j\in D}\sum_{m\in M} x_{j,2n+1,v}^{m} = 1 \qquad (v\in V), \tag{4}$$

$$\sum_{j\in N}\sum_{m\in M} x_{i,j,v}^{m} - \sum_{j\in N}\sum_{m\in M} x_{n+1,j,v}^{m} = 0 \qquad (i\in P, v\in V), \tag{5}$$

$$\sum_{j\in N}\sum_{m\in M} x_{i,j,v}^{m} - \sum_{j\in N}\sum_{m\in M} x_{i,j,v}^{m} = 0 \qquad (i\in P\cup D, v\in V), \tag{6}$$

$$t_j \ge t_i + S_i + c_{i,j}^{m} - B\left(1 - x_{i,j,m}^{v}\right) \qquad (i\in N, j\in N, m\in M, v\in V), \tag{7}$$

$$t_i - T_{i,j}^{m-1} x_{i,j,v}^{m} \ge 0 \qquad (i\in N, j\in N, m\in M, v\in V), \tag{8}$$

$$t_i + B x_{i,j,v}^{m} \le T_{i,j}^{m} + B \qquad (i\in N, j\in N, m\in M, v\in V), \tag{9}$$

$$e_i \le t_i \le l_i \qquad (i\in N, v\in V), \tag{10}$$

$$DRT_i \le t_{n+i} - t_i \le MRT_i \qquad (i \in P), \qquad (11)$$

$$w_j \ge w_i + q_i - B\left(1 - \sum_{m \in M} x_{i,j,v}^m\right) \qquad (i \in N, j \in N, v \in V), \qquad (12)$$

$$w_i \le Q_v \qquad (i \in P \cup D, v \in V), \qquad (13)$$

$$x_{i,j,m}^v \in \{0,1\} \qquad , \qquad (14)$$

$$m \in M \qquad , \qquad (15)$$

$$v \in V \qquad , \qquad (16)$$

$$t_i \ge 0 \qquad , \qquad (17)$$

$$w_i \ge 0 \qquad , \qquad (18)$$

$$N \in \{\{0\} \cup \{2n+1\} \cup P \cup D\} \qquad (19)$$

Table 1. The notations of the time-dependent DARP formulation

Notation	Contents
N	Node set (n = 0, 2n+1:Depot; n=1,…,n: Pick-up nodes; n=n+1,…,2n: Delivery nodes),
P	Pick-up nodes, P={1,…,n},
D	Delivery nodes, D={n+1,…,2n},
V	Vehicle set,
m	Number of time interval,
B	A large number,
$c_{i,j}^m$	Travel time from node i to node j at the time interval m,
S_i	The service time at node i,
Q_v	The capacity of vehicle v,
q_i	The load at node i,
$T_{i,j}^m$	Upper bound for time interval m for link(i, j),
e_i	Earliest time that the vehicle can arrive at node i,
l_i	Latest time that the vehicle can arrive at node i,
t_i	The time starts service at node i,
w_i	The load of vehicle upon leaving node i,
DPT_i	Desired pickup time at node i $(i \in P)$,
DDT_i	Desired delivery time at node i $(i \in D)$,
DRT_i	Direct ride time $(DRT_i = t_{i+n} - t_i - S_i)$,
MRT_i	Maximum ride time,
$x_{i,j,v}^m$	If any vehicle v travels from node i to node j during the interval m, the variable is equal to 1. Otherwise is equal to 0.

The objective function (1) is to minimize the total travel time. Constraints (2) ensure that each demand must be served precisely once and each demand is only allowed to be visited by one vehicle. Constraints (3) and (4) ensure that all vehicle shave to start from the depot and return to the depot. Constraints (5) are precedence constraints. Precedence constraints mean that each customer must pick up first and then delivery in the same vehicle.

Constraints (6) are the flow conservation equations. Constraints (7) calculate the departure time to node j. Constraints (8) and (9) are the temporal constraints. If the vehicle travels from demand i to demand j during time interval m, the departure time of the vehicle from node i is between upper bound for time interval m-1 and upper bound for time interval m. Constraints (10) impose the time windows restrictions. Constraints (11) ensure that the ride time of each customer i must be between directed ride time and maximum ride time. Constraints (12) and (13) impose the capacity constraints. Constraints (12) are subtour elimination constraints (Parragh et al., 2008). Constraints (13) ensure that all vehicles not exceed the vehicle capacity limitation.

Based on time-dependent DARP formulation, the time-dependent DARP formulation can be reformulated using path flow instead of link flows as a set partitioning problem. The Dantzig-Wolfe decomposition technique is applied to decompose time-dependent DARP formulation into a master problem (set portioning problem) and a subproblem (constrained shortest path problem). The detailed formulation and discussion for a master problem and a subproblem are stated as follows:

Minimize

$$\sum_{r \in R} c_r y_r \tag{20}$$

subject to

$$\sum_{r \in R} a_{ir} y_r = 1 \qquad (i \in P), \tag{21}$$

$$y_r \in \{0,1\} \qquad (r \in R), \tag{22}$$

Table 2. The notations of a master problem

Variables	Contents
R	The set of all feasible routes satisfying constraints (2)~(19),
c_r	The cost of the route r,
c_{ij}	The cost of the link (i, j).
a_{ir}	The number of times node is visited by route r $(i \in P)$,
y_r	If the route r is used in the solution, the variable is equal to 1. Otherwise is equal to 0.
π_i	The dual variables associated with the set partitioning problem constraints (20).

The objective function (20) is to minimize total cost of the selected route. Constraints (21) ensure that each request is visited by one vehicle. In this paper, the mathematical model defined by constraints (20) to (22) refers to as master problem. To solve the set partitioning formulation easily, the constraints (22) can be relaxed to constraints (23).

$$y_r \geq 0 \qquad (r \in R), \tag{23}$$

In practice, the feasible routes are difficult to enumerate when the instance is a large scale problem. In this paper, the column generation algorithm is proposed to solve the master problem. The important concept of the column generation is never to enumerate all possible feasible routes for the formulation. Each column generate by a subproblem which is refer to as a constrained shortest path problem. The subproblem is defined as follows:

Minimize

$$\sum_{i,j \in N} \left(c_{ij} - \pi_i\right) x_{ij} \tag{24}$$

Subject to constraints (2)~(19).

4. SOLUTION ALGORITHM

The process of the solution algorithm is depicted in Figure 2. The details of each step are described as follows:

1. Input Data: Three principal input data are necessary including time-dependent travel time matrices, vehicle data and demand data. In this paper, the traffic simulation software, DynaTAIWAN (Hu, et al., 2007) is applied to generate the time-dependent travel time matrices according to the network geometric data and OD demand data.
2. DARP Formulation: Based on the necessary input data, the time-dependent DARP with time window is formulated.
3. Decomposition: The Dantzig-Wolfe decomposition is applied to decompose time-dependent DARP with time window into master problem (set portioning problem) and subproblem (constrained shortest path problem).
4. Column Generation: In the process of the column generation approach, the first is to generate initial feasible solution. In this paper, the initial feasible solution is to assign vehicles to visit customers. Each vehicle only serves one customers. According to the initial feasible solution, the formulation of the master problem is constructed and obtains multipliers (dual variables) to feed into subproblem. Based on the multipliers, the subproblem takes advantage of dynamic programming approach to find the optimal column with the minimum reduced cost. If the reduced cost of the optimal column is less than 0, this column will feed into master problem and re-optimize the master problem to generate new multipliers. Otherwise, the solution process will go next step.
5. Branch and bound: If the final solution is not integer solution, the branch and bound is used to ensure that final result is integer.
6. Output: After the solution process terminated, the vehicle routes and service sequences are obtained.

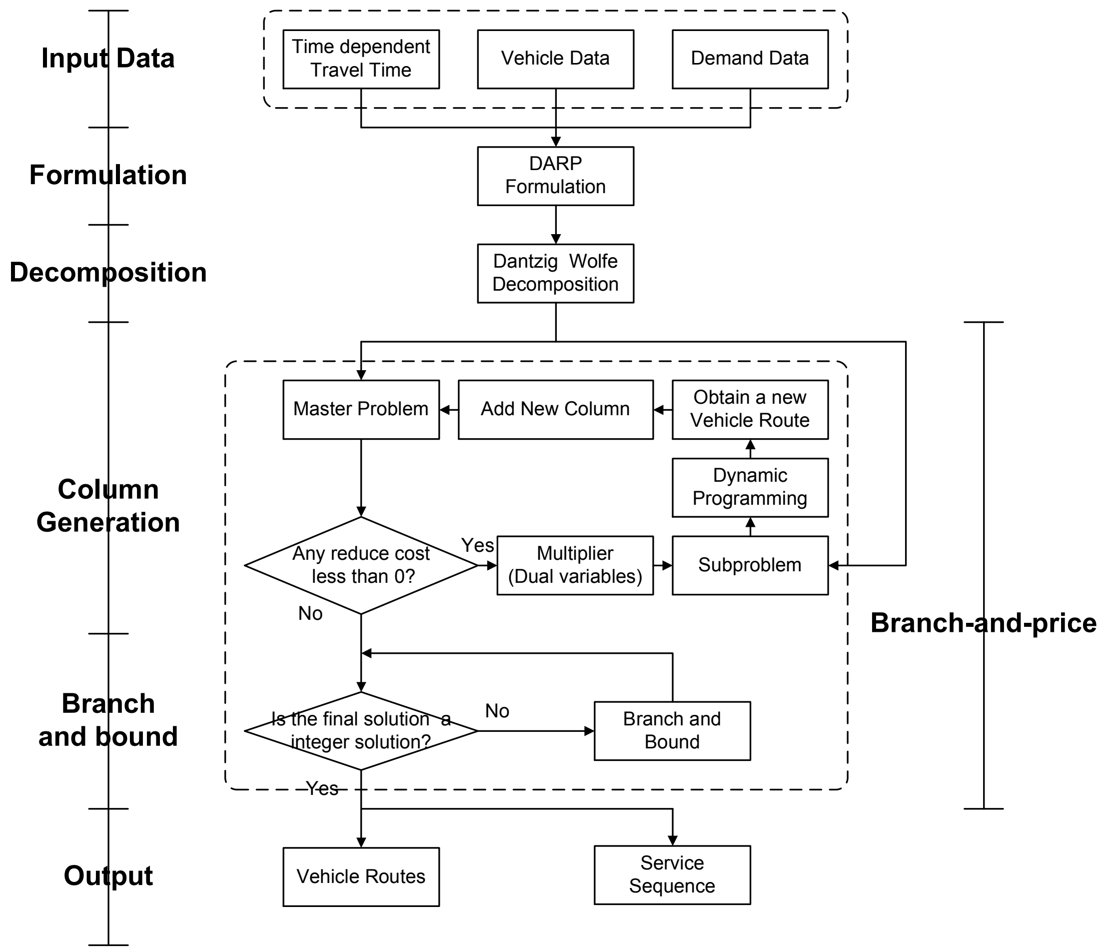

Figure 2. Solution Algorithm

5. NUMERICAL EXPERIMENTS

Numerical experiments are conducted on a sub-network of Kaohsiung City, Taiwan (see in Figure 3). This network includes 132 nodes and 363 links. The experimental parameters are specified as follows: the maximum simulation time is 600 minutes; the warm-up time is 20 minutes; the instance in the numerical experiments includes 2 depot nodes (source node and sink node), 10 pickup nodes and 10 delivery nodes; the number of time intervals is 7 (m=7) and each time interval has same length of time (5 min). Each vehicle can accommodate 4 persons. Each customer has one pickup node, corresponding delivery node, time window constant and maximum ride time limitation. In the experimental instances, all customers are DPT-specified customers (see in Figure 1). The expected pickup service time is equal to earliest pickup time (EPT) and the expected delivery time service time is equal to earliest delivery time (EDT). The expected pickup service time of each customer is randomly generated. During the vehicle routing, if the actual vehicle arrival time is less than earliest pickup time, the vehicle can wait 5 minutes at the pickup node until the earliest pickup time. The maximum ride time is the function of the directed ride. The detail of this function is shown as follow:

$$MRT_i = a + b \cdot DRT_i \tag{25}$$

In the numerical experiments, the coefficient a and b of the function (25) are set 10 and 2 respectively.

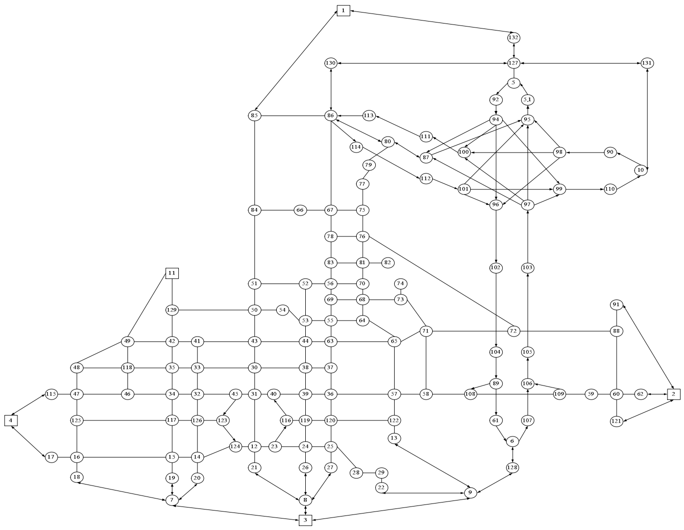

Figure 3. A sub-network of Kaohsiung City, Taiwan

The numerical experiments consider two main experimental factors including time window and traffic conditions. In the parameter setting of the time window, the length of the time window including 5, 10, 15, 20, 25, 30 (min) are considered. Three different kinds of traffic conditions are considered incorporating light traffic condition, medium traffic condition and heavy traffic condition. The basic results of traffic simulation are shown is table 3. In the experiments, the scenario “H_10” means that the experiment is in the heavy traffic condition and the length of time window is 10.

Table 3. Basic results of traffic simulation

Scenario	Number of Vehicle	Number of Motorcycle	Average Travel Time(min)	Average Stopped Time(min)
Light Traffic	20760	12693	8.72	3.46
Medium Traffic	43456	31859	20.24	12.22
Heavy Traffic	64227	44561	47.60	29.46

In the numerical experiments, average pickup delay time and average delivery delay time are used to evaluate the customer satisfaction. The definitions of the pickup delay time and delivery delay time are specified as follows:

$$Average\ pickup\ delay\ time = \frac{\sum_{i \in P}\left(Actual\ \ pickup\ time - Expected\ \ pickup\ time\right)}{The\ number\ of\ \ customers} \qquad (26)$$

$$Average\,delivery\,delay\,time = \frac{\sum_{i \in P}\left(Actual\,delivery\,time - Expected\,delivery\,time\right)}{The\,number\,of\;customers} \quad (27)$$

The details of the vehicle routes and the travel time for each vehicle under different traffic scenarios are listed in Table 4, Table5 and Table 6. The objective value, computational time (CPU time), average pickup delay time, average delivery delay time, number of vehicles and average travel time in different traffic scenarios are summarized in Table 7.

Table 4. The vehicle routes and travel time under the light traffic condition

Scenario	Vehicle ID (Route)	Travel Time (min)
L_5	1(0, 3, 1, 13, 11, 21)	44.68
	2(0, 10, 2, 20, 12, 21)	44.76
	3(0, 5, 4, 15, 14, 21)	42.64
	4(0, 6, 7, 16, 17, 21)	43.30
	5(0, 8, 18, 9, 19, 21)	40.67
L_10	1(0, 5, 15, 21)	26.43
	2(0, 6, 16, 21)	19.06
	3(0, 7, 9, 19, 17, 21)	40.57
	4(0, 8, 3, 1, 18, 13, 11, 21)	57.26
	5(0, 10, 2, 4, 14, 20, 12, 21)	59.41
L_15	1(0, 10, 3, 1, 2, 13, 20, 12, 11, 21)	72.66
	2(0, 6, 7, 16, 9, 19, 17, 21)	53.71
	3(0, 8, 18, 5, 4, 15, 14, 21)	56.53
L_20	1(0, 10, 1, 3, 2, 13, 12, 20, 11, 21)	69.81
	2(0, 6, 7, 16, 9, 19, 17, 21)	53.71
	3(0, 8, 18, 5, 4, 15, 14, 21)	56.53
L_25	1(0, 5, 15, 21)	26.43
	2(0, 6, 16, 21)	19.06
	3(0, 10, 1, 3, 2, 13, 11, 20, 12, 21)	69.79
	4(0, 8, 7, 18, 4, 9, 19, 17, 14, 21)	71.56
L_30	1(0, 3, 13, 21)	29.00
	2(0, 6, 16, 21)	19.06
	3(0, 8, 18, 10, 4, 1, 2, 12, 20, 11, 14, 21)	82.32
	4(0, 7, 5, 9, 19, 17, 15, 21)	56.20

Table 5. The vehicle routes and travel time under the medium traffic condition

Scenario	Vehicle ID (Route)	Travel Time (min)
M_5	1(0, 3, 1, 13, 11, 21)	49.13
	2(0, 10, 2, 12, 20, 21)	47.17
	3(0, 6, 7, 16, 17, 21)	46.15
	4(0, 5, 4, 15, 14, 21)	45.99
	5(0, 8, 18, 9, 19, 21)	45.39
M_10	1(0, 6, 16, 21)	20.30
	2(0, 3, 1, 2, 13, 12, 11, 21)	61.41
	3(0, 7, 9, 19, 17, 21)	42.73
	4(0, 10, 5, 15, 20, 21)	50.44
	5(0, 8, 4, 18, 14, 21)	42.42

M_15	1(0, 5, 15, 21)	28.71
	2(0, 10, 1, 2, 12, 20, 11, 21)	60.15
	3(0, 8, 3, 4, 18, 13, 14, 21)	58.74
	4(0, 6, 7, 16, 9, 19, 17, 21)	56.56
M_20	1(0, 10, 1, 3, 2, 13, 20, 12, 11, 21)	72.20
	2(0, 6, 7, 16, 9, 19, 17, 21)	56.56
	3(0, 8, 18, 5, 4, 15, 14, 21)	62.29
M_25	1(0, 5, 15, 21)	28.71
	2(0, 6, 16, 21)	20.30
	3(0, 10, 1, 3, 2, 13, 12, 20, 11, 21)	71.64
	4(0, 8, 7, 18, 9, 4, 19, 17, 14, 21)	76.98
M_30	1(0, 6, 16, 21)	20.30
	2(0, 9, 19, 21)	28.46
	3(0, 10, 1, 3, 2, 13, 12, 20, 11, 21)	71.64
	4(0, 8, 7, 18, 5, 4, 17, 15, 14, 21)	79.56

Table 6. The vehicle routes and travel time under the heavy traffic condition

Scenario	Vehicle ID (Route)	Travel Time (min)
H_5	1(0, 10, 1, 20, 11, 21)	52.21
	2(0, 3, 2, 13, 12, 21)	48.06
	3(0, 5, 4, 15, 14, 21)	51.05
	4(0, 6, 7, 16, 17, 21)	48.70
	5(0, 8, 18, 9, 19, 21)	48.07
H_10	1(0, 6, 16, 21)	21.79
	2(0, 3, 1, 2, 13, 12, 11, 21)	62.51
	3(0, 7, 9, 19, 17, 21)	45.24
	4(0, 10, 5, 15, 20, 21)	54.48
	5(0, 8, 4, 18, 14, 21)	44.47
H_15	1(0, 5, 15, 21)	32.24
	2(0, 3, 1, 2, 13, 12, 11, 21)	62.51
	3(0, 6, 7, 16, 9, 19, 17, 21)	59.08
	4(0, 8, 10, 4, 18, 20, 14, 21)	66.37
H_20	1(0, 10, 1, 3, 2, 13, 20, 12, 11, 21)	74.45
	2(0, 8, 18, 5, 4, 15, 14, 21)	67.24
	3(0, 6, 7, 16, 9, 19, 17, 21)	59.08
H_25	1(0, 3, 4, 1, 2, 13, 12, 11, 14, 21)	77.37
	2(0, 6, 7, 16, 9, 19, 17, 21)	59.08
	3(0, 10, 5, 8, 15, 18, 20, 21)	70.08
H_30	1(0, 5, 15, 21)	32.24
	2(0, 8, 18, 21)	26.26
	3(0, 10, 1, 3, 2, 13, 11, 20, 12, 21)	73.87
	4(0, 6, 7, 16, 9, 4, 19, 17, 14, 21)	76.64

Table 7. The results under the different traffic conditions

Scenario	Objective (min)	CPU Time (sec)	Average Pickup Delay Time (min)	Average Delivery Delay Time (min)	Number of vehicles	Average travel time (min)
L_5	216.05	0.73	1.34	9.80	5	43.21
L_10	202.73	1.90	2.07	17.06	5	40.55

L_15	182.90	9.20	5.60	22.77	3	60.97
L_20	180.05	31.25	6.27	23.06	3	60.02
L_25	186.84	110.71	6.64	24.92	4	46.71
L_30	186.58	430.52	9.48	25.96	4	46.65
M_5	233.83	0.76	1.29	12.26	5	46.77
M_10	217.30	1.28	2.99	15.39	5	43.46
M_15	204.16	4.71	3.64	20.78	4	51.04
M_20	191.05	17.30	7.02	24.80	3	63.68
M_25	197.63	45.62	7.08	26.00	4	49.41
M_30	199.96	176.77	8.15	27.58	4	49.99
H_5	248.09	0.61	1.62	13.58	5	49.62
H_10	228.49	1.17	3.25	16.35	5	45.70
H_15	220.20	3.84	4.61	22.54	4	55.05
H_20	200.77	14.30	7.57	25.93	3	66.92
H_25	206.53	28.05	6.44	26.82	3	68.84
H_30	209.01	97.84	7.41	26.57	4	52.25

The relationship between the objective value and the length of the time window is shown in Figure 4. Basically, the experimental results have the same patterns in the three different traffic scenarios. The results indicate that the objective value increases as the length of the time window reduces. One of the reasons might be that the tight time window constraints will reduce the region of the feasible solutions. In additional, the objective value in the heavy traffic scenario is higher than other traffic scenarios.

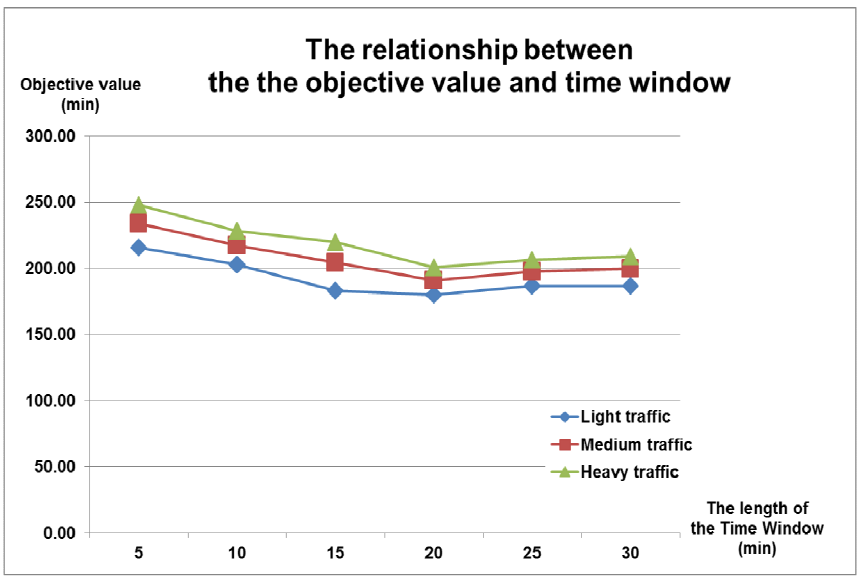

Figure 4. The relationship between the objective value and time window

The relationship between the computational time and the length of the time window is depicted in Figure 5. The results reveal that the length of the time window dominates the computational time. When the time window constraints loosen, the computational time will increase exponentially.

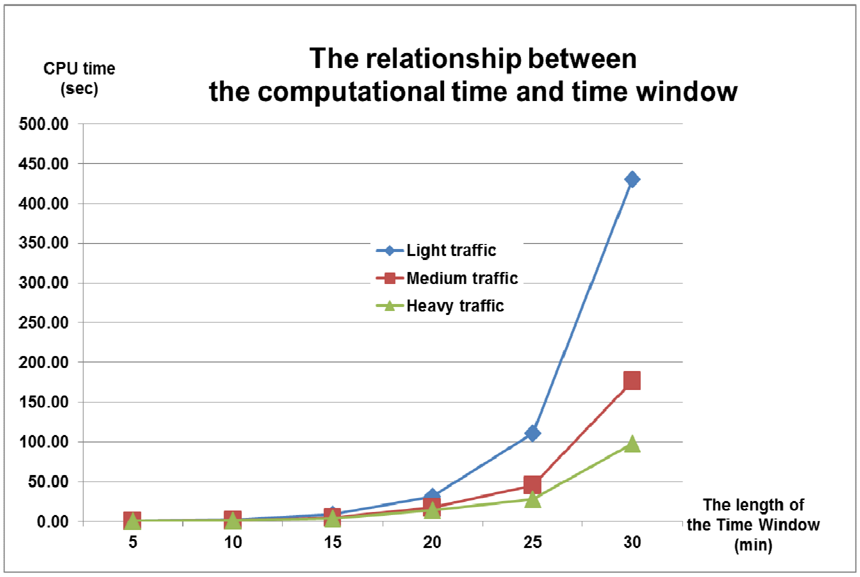

Figure 5. The relationship between the computational time and time window

The average pickup delay time and average delivery delay time are shown in Figure 6 and Figure 7 respectively. The results reveal that the experimental scenarios with tight time window will obtain the vehicle routes with less average pickup delay time and average delivery delay time.

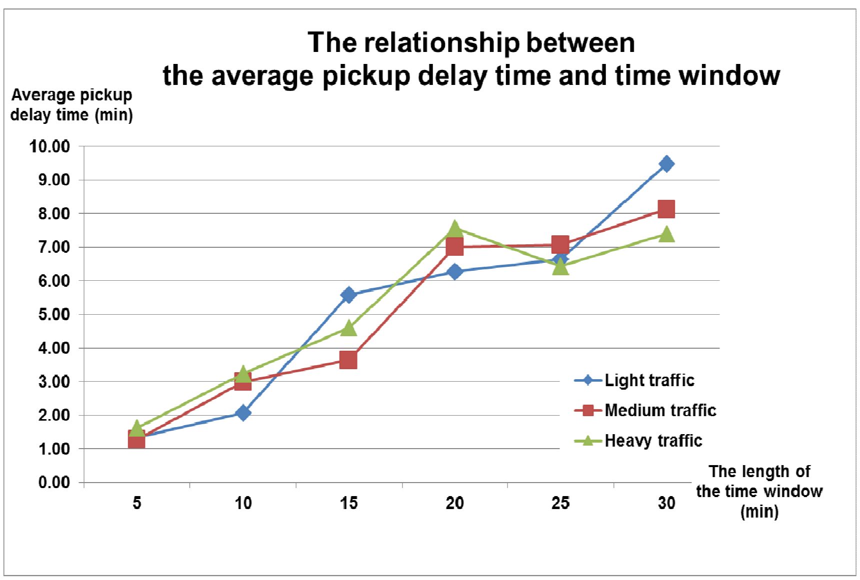

Figure 6. The relationship between the average pickup delay time and time window

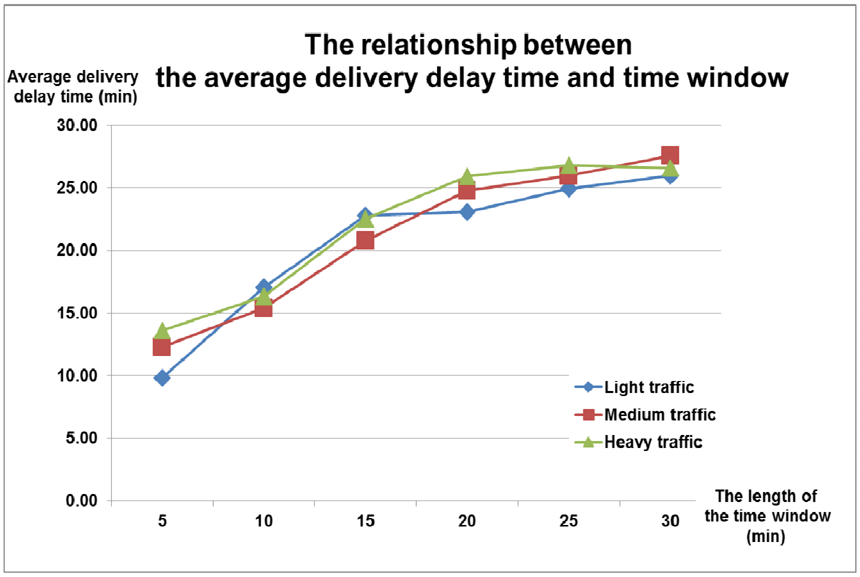

Figure 7. The relationship between the average delivery delay time and time window

The relationship between the number of vehicles and the length of the time window is depicted in Figure 8. The results indicate that the number of vehicles decreases slightly as the length of the time window increases. One of the reasons might be that the scenario under the wide length of the time window can easily assign more customers into one vehicle.

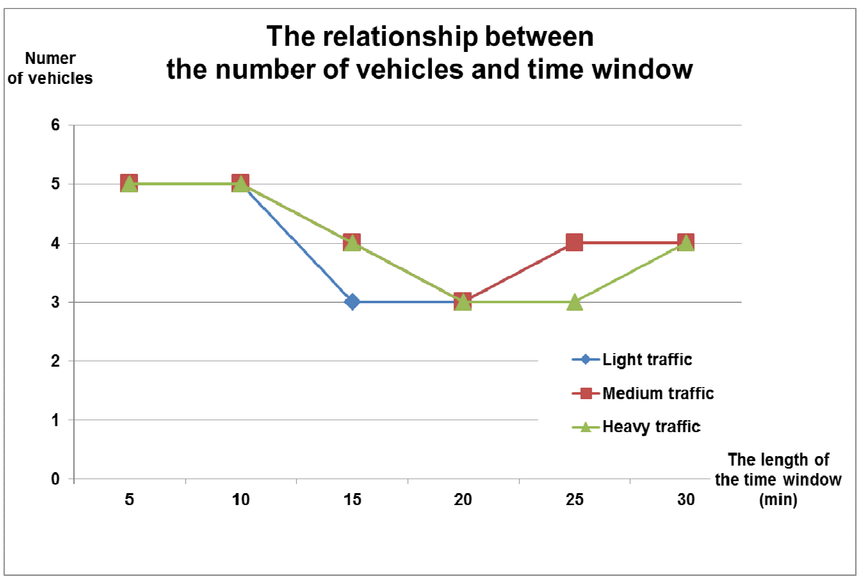

Figure 8. The relationship between the number of vehicles and time window

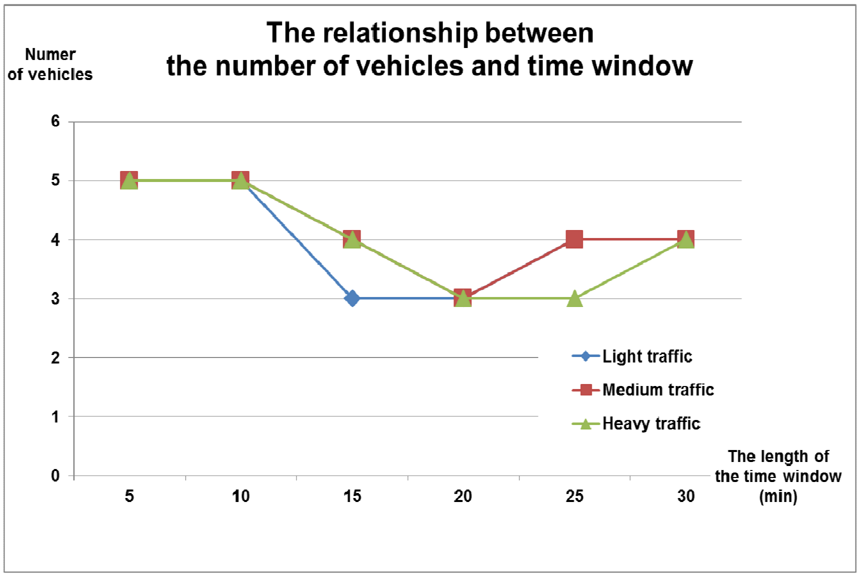

Figure 9. The relationship between the average travel time and time window

The relationship between the average travel time and the length of the time window is shown in Figure 9. The results reveal that the variation of the time window cannot significantly affect the average travel time for each scenario.

6. CONCLUSIONS

In this paper, the time-dependent DARP with time window is formulated. Based on the time-dependent DARP formulation, the Dantzig-Wolfe decomposition is used to decompose the DARP into the master problem and subproblem. Then, the brand-and-price approach is applied to deal with the time-dependent DARP with time window. According to the results of the numerical experiments, the brand-and-price approach is capable of handling DARP.

In the numerical experiments, two major factors including the time window and traffic condition are experimented to implement the sensitivity analysis. The experimental results show that the length of the time window can significantly affect the objective value, computational time, average pickup delay time, average delivery delay time and number of vehicles. When the length of the time window increase, the objective value will decrease slightly and the computational time will increase exponentially.

REFERENCES

Cordeau, J. -F. and Laporte, G. (2003) The Dial-a-Ride Problem (DARP):Variants, modeling issues and algorithms, Quarterly Journal of the Belgian, French and Italian Operations Research Societies, Vol. 1, No.2, 89-101.

Cordeau, J. -F., and Laporte, G. (2003) A tabu search heuristic for the static multi-vehicle dial-a-ride problem, Transportation Research Part B, Vol. 37, No. 6, 579–594.

Cordeau, J. -F. (2006) A Branch-and-cut algorithm for the dial-a-ride problem, Operations Research, Vol. 54, No. 3, 573–586.

Cordeau, J. -F. and Laporte, G. (2007) The dial-a-ride problem: models and algorithms, Annals of Operations Research, Vol. 153, 29-46.

Cortes, C.E., Saez, D., Nunez, A. and Muñoz-Carpintero, D. (2009) Hybrid Adaptive

Predictive Control for a Dynamic Pickup and Delivery Problem, Transportation Science, Vol. 43, No. 1, 27-42.

Cortes, C.E., Matamala, M. and Contardo, C. (2010) The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method, European Journal of Operational Research, Vol. 200, No. 3, 711-724.

Diana, M. and Dessouky, M. M. (2004) A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows, Transportation Research Part B, Vol. 38, No.6, 539-557.

Dumas, Y., Desrosiers, J. and Soumis, F. (1991) The pickup and delivery problem with time windows, European Journal of Operational Research, Vol. 54, No. 1, 7-22.

D'Souza, C., Omkar, S. N. and Senthilnath, J. (2012) Pickup and Delivery Problem Using Metaheuristics Techniques, Expert Systems with Applications, Vol. 39, No. 1, 328-334.

Fabri, A. and Recht, P. (2006) On dynamic pickup and delivery vehicle routing with several time windows and waiting times, Transportation Research Part B, Vol. 40, No. 4, 335-350.

Fu, L. (2002) Scheduling dial-a-ride paratransit under time-varying, stochastic congestion, Transportation Research Part B, Vol. 36, No. 6, 485-506.

Hu, T.Y., Liao, T.Y., Chen, L.W., Huang, Y.K., and Chiang, M.L. (2007) Dynamic simulation-assignment model (DynaTAIWAN) under mixed traffic flows for ITS application, In: Proceedings of 86th Transportation Research Board Annual Meeting, Washington, D.C., U.S.A.

Hu, T. Y. and Chang, C. P. (2011) Time-Dependent Dial-a-Ride Problems: Formulation Development and Numerical Experiments, Journal of the Eastern Asia Society for Transportation Studies, Vol. 9, 690-701.

Jaw, J.-J., Odoni, A.R., Psarafits, H.N. and Wilson, N.H.M. (1986) A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows, Transportation Research Part B, Vol. 20, No. 3, 243-257.

Jørgensen, R. M., Larsen, J., and Bergvinsdottir, K. B. (2007) Solving the dial-a-ride problem using genetic algorithms, Journal of the Operational Research Society, Vol. 58, No. 10, 1321-1331.

Kim, T. and Haghani, A. (2011) Model and Algorithm Considering Time-Varying Travel Times to Solve Static Multidepot Dial-a-Ride Problem, In: Proceedings of 87th Transportation Research Board Annual Meeting, Washington D.C., U.S.A.

Li, H. and Lim, A. (2001) A MetaHeuristic for the Pickup and Delivery Problem with Time Windows, In: Proceedings of the 13th International Conference on Tools with Artificial Intelligence, Dallas, Texas, U.S.A.

Lois, A., Ziliaskopoulos, A. K. and Aifantopoulou, G (2007), A Very Large Scale Neighborhood Heuristic Algorithm for the Multivehicle Dial a Ride with Time Windows, In: Proceedings of 87th Transportation Research Board Annual Meeting, Washington D.C., U.S.A.

Malandraki, C. and Daskin, MS. (1992) Time Dependent Vehicle Routing Problems: Formulations, Properties and Heuristic Algorithms, Transportation Science, Vol. 26, No.3, 185-199.

Parragh, S.N., Doerner, K.F. and Hartl, R.F. (2008) A survey on pickup and delivery problems. Part II: transportation between pickup and delivery locations, Journal fur Betriebswirtschaft, Vol. 58, No. 2, 81–117.

Psaraftis, H. N. (1980) A dynamic programming approach to the single-vehicle, many-to-many immediate request dial-a-ride problem, Transportation Science, Vol. 14, No. 2, 130–154.

Psarafits, H. N. (1983) Analysis of an O(N2) Heuristic for Many-to-many Euclidean

Dial-a-ride Problem, Transportation Research Part B, Vol. 17, No. 2, 133-145.
Ropke, S. and Cordeau, J. –F. (2009) Branch and Cut and Price for the Pickup and Delivery Problem with Time Windows, Transportation Science, Vol.43, No. 3, 267-286.
Ropke, S. and Pisinger, D. (2006) An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows, Transportation Science, Vol. 40, No. 4, 455-472.
Wong, K. I., and Bell, M. G. H. (2006) Solution of the dial-a-ride problem with multi-dimensional capacity constraints, International Transactions in Operational Research, Vol. 13, No. 3, 195–208.
Xiang, Z., Chu, C., and Chen, H. (2006) A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints, European Journal of Operational Research, Vol. 174, No. 2, 1117–1139.