Public Transp (2009) 1: 39–54
DOI 10.1007/s12469-008-0006-1

ORIGINAL PAPER

The Integrated Dial-a-Ride Problem

Carl H. Häll · Henrik Andersson ·
Jan T. Lundgren · Peter Värbrand

Published online: 8 November 2008

Abstract In this paper we present a formulation of the Integrated Dial-a-Ride (IDARP). This problem is to schedule dial-a-ride requests, where some part of each journey may be carried out by a fixed route service. The IDARP is a generalization of the Dial-a-Ride Problem. An arc-based formulation is proposed, and it is shown how the model can be made easier to solve by arc elimination, variable substitution and the introduction of subtour elimination constraints. Small instances of the IDARP can be solved using an exact solution method, and one such instance is studied. We also describe how input and output data can be created and visualized in a geographic information system.

1 Introduction

Public transportation systems are evolving towards more flexible solutions in order to better serve the mobility needs of the population and to reduce the cost of transportation. For transportation of elderly and disabled people this development is particularly important. Today, most transportation services for these user groups are demand responsive services of a door-to-door type. Given a set of requests for travel, including individual passengers or groups of passengers, with specified origins and destinations, the planning problem is to design routes and schedules for the fleet of vehicles. The schedule has to satisfy passengers' requirements and desires in terms of user inconvenience, e.g., stated as waiting times, travel times or deviations from desired

C.H. Häll (✉) · J.T. Lundgren · P. Värbrand
Department of Science and Technology (ITN), Linköping University, 60174 Norrköping, Sweden
e-mail: carha@itn.liu.se

H. Andersson
Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, 7491 Trondheim, Norway

departure and arrival times. The overall goal is to balance the user inconvenience against operating costs of the vehicles. This problem is known as the Dial-a-Ride Problem (DARP) and it is a special case of the pick-up and delivery problem.

This paper considers a form of flexible public transportation system where a demand responsive service is integrated with a fixed route service. This type of system can use the already existing fixed route service for the major part of the journeys and thereby only needs to use the more expensive demand responsive service for shorter distances. In this way municipalities can significantly reduce their costs for special needs permits for elderly and disabled by increasing the utilization of the fleet or reducing the number of vehicles. The problem for many of those using dial-a-ride services is often not to ride on the regular public transport, but to take themselves to, and from, the public transport. The increased accessibility to the public transport service can help some parts of this user group to cope without the permits.

This form of integrated service exists in some areas of Sweden but handles quite few requests, and mainly considers longer journeys (e.g., as described in Börjesson 2001). Requests suitable to be served as a combination of dial-a-ride and fixed route service are identified and how the combination should be made is decided manually. The demand responsive journey-legs are then planned using the same computerized tool as all dial-a-ride requests that are not considered to be served by the integrated service. To take this partially manual planning process a step further, we present a model for the planning of the integrated service.

The Integrated Dial-a-Ride Problem (IDARP) is to design vehicle routes and schedules for a dial-a-ride service where some part of each request may be performed by a fixed route service. The design of the fixed routes is assumed to be given and the question is how to use the existing fixed routes in an optimal way for the demand responsive service.

The general IDARP is formulated using an explicit network optimization model. For practical applications, the problem size in terms of number of nodes and arcs becomes very large, and it is necessary to find a way to reduce the model size. Different ways to make the model easier to solve, including arc elimination, variable substitution and subtour elimination, are presented. Small problem instances have been solved using an exact solution method and we believe that with some future developments of the method, also medium sized problems can be solved exactly.

The outline of the paper is as follows. Section 2 is a literature review presenting earlier work on both the IDARP and the DARP. In Sect. 3 our exact formulation of the IDARP is presented and Sect. 4 describes different steps taken to make the problem easier to solve. Section 5 includes a numerical example and describes how input and output data can be handled in a geographic information system (GIS). Finally in Sect. 6 some conclusions are drawn.

2 Literature review

The first work on integrated transportation problems was presented by Wilson et al. (1976). The problem has a passenger utility function as its objective, and this function is maximized subject to a series of level of service constraints. This means that the costs of the operator are not included explicitly in the model. A trip insertion heuristic is used to schedule both passenger and vehicle trips. Liaw et al. (1996) formulate a model for the integrated problem. Heuristics to solve the problem are also presented. Both static and dynamic instances are solved. In Hickman and Blume (2001) passenger trips are scheduled on-line (dynamically), and the vehicle itineraries are scheduled off-line (static) after all passenger trips have been scheduled. Both transit agency cost and passenger level of service are explicitly included in the proposed method. In Aldaihani and Dessouky (2003), two different transportation modes, a demand responsive (curb-to-curb) service and a fixed route service, are integrated in order to satisfy a given demand. The objective function contains two measures of performance, minimize total travel distance of demand responsive vehicles and minimize total travel time of passengers. The proposed heuristic is tested on real-life data obtained from Antelope Valley Transit Authority. Horn (2004) describes procedures for planning journeys combining fixed route and demand responsive modes. The tests made with simulated demand shows that the procedures are well suited for a real-time traveler information system.

Since the DARP is the basis for the IDARP, contributions solving the DARP can be valuable when studying the IDARP. Both problems are NP-hard, and therefore most recent work focus on heuristic methods. However, Cordeau (2006) and Ropke et al. (2007) describe exact formulations of the DARP as well as branch-and-cut algorithms. In addition, the authors also describe several families of valid inequalities, especially the family of subtour elimination constraints.

Some early works on the DARP are Wilson et al. (1971), Psaraftis (1980, 1983). Wilson et al. (1971) investigate the dynamic DARP. Psaraftis (1980) solved both the static and the dynamic version of the single vehicle DARP. In Psaraftis (1983), the problem is extended to also include time windows. Jaw et al. (1986) continued this work by developing a heuristic algorithm to solve a static case of the multi vehicle DARP.

Regarding metaheuristics for solving the DARP, tabu search is most commonly used. In Cordeau and Laporte (2003), the authors use a tabu search algorithm on several different data sets. In Attanasio et al. (2004), different parallel heuristics based on tabu search are tested for the dynamic DARP. The authors use the static tabu search developed by Cordeau and Laporte (2003) to find a solution to the static problem of the requests known at the start of the planning horizon.

Other techniques have also been used. Borndörfer et al. (1997) use a set partitioning approach consisting of a clustering step and a chaining step, both solved by a branch-and-cut algorithm. Baugh et al. (1998) use simulated annealing to cluster the customers and a nearest neighbor heuristic to find the routes within the clusters. Genetic algorithms are used by Uchimura et al. (2002) and Bergvinsdottir et al. (2004). Uchimura et al. (2002) focus on comparison between the genetic algorithm, a standard edge exchange algorithm (2-opt algorithm) and a combination of the genetic algorithm and the 2-opt. Bergvinsdottir et al. (2004) present a genetic algorithm based on a cluster-first route-second approach. Also in Chan (2004), a cluster-first route-second approach is used, where the clustering is made both with tabu search as well as with scatter search. Madsen et al. (1995) use an algorithm based on an insertion heuristic to solve the DARP with multiple capacities and multiple objectives. The algorithm is developed to solve a real-life problem of scheduling transportation for elderly and disabled in Copenhagen, Denmark. Ioachim et al. (1995) describe a method based on clustering. Experiments are made on instances of 50 to 250 requests, as well as on a real-life problem consisting of 2545 requests. Diana and Dessouky (2004) present a parallel regret insertion heuristic to solve large instances of the DARP. Data sets of 500 and 1000 requests are tested. Toth and Vigo (1997) develop a parallel insertion heuristic to find good solutions also to large instances within quite small computational times.

3 Model formulation

A number of requested journeys are to be scheduled. Each request has a specified pick-up location (node) and a drop-off location (node), as well as a time window specified for one of these nodes, or both. Furthermore, each request also has a maximum ride time. The requests can be carried out by a dial-a-ride service, but some part of each journey may be carried out by a fixed route service. This means that the passenger can either be carried by the dial-a-ride service from the pick-up node to the drop-off node without changing vehicles, or from the pick-up node to a transfer node, where the passenger can change to a fixed route service. If the passenger is carried to a transfer node, it uses the fixed route to travel to another transfer node where another vehicle in the fleet of dial-a-ride vehicles picks up the passenger and carries the passenger to its drop-off node. Each request contains one or several passengers and requires a certain capacity in a vehicle, for the persons and any wheelchairs, walkers, luggage etc. This needed capacity generates a load at each node describing the capacity required if the request is picked up at the node, or made available if the request is dropped off. The fleet of dial-a-ride vehicles is assumed to be homogeneous. All vehicles have the same capacity, the same maximum duration time (time from when the vehicle leaves the depot until it must return to the depot) and are stationed at the same depot. The only difference between the vehicles is the time window in which each vehicle can leave the depot.

The formulation has an underlying network structure. This problem is based on the DARP, and therefore we have based our IDARP formulation on the directed graph formulation of the DARP, described in Cordeau (2006). The main expansion of the original DARP-formulation is that in the IDARP both vehicle itineraries and customer itineraries are scheduled.

In the directed graph $G = (N, A)$, N is the set of all nodes, including pick-up nodes, drop-off nodes, depot nodes and transfer nodes, and A is the set of arcs connecting the nodes. Each arc $(i, j) \in A$ has an associated cost c_{ij} and travel time t_{ij}. Among all nodes in N, the subset $P = \{1, \ldots, n\}$ contains the pick-up nodes for the n requests, and $D = \{n + 1, \ldots, 2n\}$ is the corresponding set of drop-off nodes. In this way node $n + i$ is the drop-off node associated with pick-up node i. The number of fixed route stops is g, and for each such stop there is a separate transfer node for each request. The total number of transfer nodes is therefore equal to the number of fixed route stops times the number of requests. These nodes form the subset $C = \{2n+2, \ldots, 2n+1+ng\}$ of N. The set C^r is the set of nodes in set C associated with request r. The origin depot is node 0 and the destination depot is node $2n + 1$. Finally, each node i has an associated time window $[e_i, l_i]$.

The set C makes the main difference between the formulation of the DARP and that of the IDARP. In the DARP all nodes $i \in N$ are visited, but for each request r in the IDARP at most two transfer nodes in C^r are visited. In the DARP, the same vehicle must visit both the pick-up node and the drop-off node belonging to a request, which is not the case in the IDARP.

The following notation is introduced:

Sets:

R set of requests

P set of pick-up nodes

D set of drop-off nodes

C set of transfer nodes

C^r set of transfer nodes associated with request r

N set of all nodes, pick-up, drop-off, depot and transfer nodes

K set of demand responsive vehicles

Variables:

$$x_{ij}^k = \begin{cases} 1, & \text{if vehicle } k \text{ travels from node } i \text{ to node } j \\ 0, & \text{otherwise} \end{cases}$$

$$y_{ij}^r = \begin{cases} 1, & \text{if request } r \text{ travels by dial-a-ride from node } i \text{ to node } j \\ 0, & \text{otherwise} \end{cases}$$

$$z_{ij} = \begin{cases} 1, & \text{if the fixed route between transfer nodes } i \text{ and } j \text{ is used} \\ 0, & \text{otherwise} \end{cases}$$

B^k the time at which vehicle k leaves the depot

B_i start of service at node i

D_i time from which a vehicle leaves the depot, until it arrives at node i

Parameters:

Q capacity of a dial-a-ride vehicle

T maximum duration time of a dial-a-ride vehicle

c_{ij} cost for any vehicle to travel from node i to node j

t_{ij} travel time from node i to node j

q^r load of request r

d_i service duration at node i

e_i earliest time at which service may begin at node i

l_i latest time at which service may begin at node i

e^k earliest time at which vehicle k can leave the depot

l^k latest time at which vehicle k can leave the depot

$$f_i^r = \begin{cases} 1, & \text{if node } i \text{ is the pick-up node of request } r \\ -1, & \text{if node } i \text{ is the drop-off node of request } r \\ 0, & \text{otherwise} \end{cases}$$

$\bar{L}_i$ maximum ride time of request i
M large positive number

The mathematical formulation becomes:

$$\min \sum_{i\in N} \sum_{j\in N} \sum_{k\in K} c_{ij} x_{ij}^k \tag{1}$$

Subject to:

$$\sum_{j\in N} \sum_{k\in K} x_{ij}^k = 1 \quad i \in P \cup D \tag{2}$$

$$\sum_{j\in N} x_{0j}^k = 1 \quad k \in K \tag{3}$$

$$\sum_{j\in N} x_{ji}^k - \sum_{j\in N} x_{ij}^k = 0 \quad i \in P \cup D \cup C, k \in K \tag{4}$$

$$\sum_{i\in N} x_{i,2n+1}^k = 1 \quad k \in K \tag{5}$$

$$\sum_{j\in N} y_{ij}^r + \sum_{j\in C^r} z_{ij} - \sum_{j\in N} y_{ji}^r - \sum_{j\in C^r} z_{ji} = f_i^r \quad r \in R, i \in C^r \tag{6}$$

$$\sum_{j\in N} y_{ij}^r - \sum_{j\in N} y_{ji}^r = f_i^r \quad r \in R, i \in N \backslash C^r \tag{7}$$

$$\sum_{r\in R} q^r y_{ij}^r \le Q \sum_{k\in K} x_{ij}^k \quad i \in N, j \in N \tag{8}$$

$$B_j \ge B^k + t_{0j} - M(1 - x_{0j}^k) \quad j \in N, k \in K \tag{9}$$

$$B_j \ge B_i + d_i + t_{ij} - M\left(1 - \sum_{k\in K} x_{ij}^k\right) \quad i \in N, j \in N \tag{10}$$

$$B_j \ge B_i + t_{ij} - M(1 - z_{ij}) \quad r \in R, (i, j) \in (C^r \times C^r) \tag{11}$$

$$e_i \le B_i \le l_i \quad i \in P \cup D \tag{12}$$

$$t_{i,n+i} \le B_{n+i} - (B_i + d_i) \le \bar{L}_i \quad i \in P \tag{13}$$

$$D_j \ge D_i + d_i + t_{ij} - M\left(1 - \sum_{k\in K} x_{ij}^k\right) \quad i \in N, j \in N \tag{14}$$

$$D_i + d_i + t_{i,2n+1} \le T \quad i \in N \tag{15}$$

$$e^k \le B^k \le l^k \quad k \in K \tag{16}$$

$$x_{ij}^k \in \{0, 1\} \quad i \in N, j \in N, k \in K \tag{17}$$

$$y_{ij}^r \geq 0 \quad r \in R, i \in N, j \in N \tag{18}$$

$$z_{ij} \in \{0, 1\} \quad r \in R, (i, j) \in (C^r \times C^r) \tag{19}$$

The objective is to minimize the total routing cost of the dial-a-ride vehicles. Constraint (2) says that all pick-up and drop-off nodes must be visited. Constraints (3)–(5) ensure that each vehicle starts at the depot, leaves each node it arrives at and ends at the depot. Constraints (6)–(7) describe the node balance of the requests. In constraint (6), describing the transfer nodes, passengers can arrive and depart in either a dial-a-ride vehicle or by a fixed route. In constraint (7) only the dial-a-ride vehicles have to be considered. Constraint (8) ensures that the sum of the passengers in all requests traveling between node i and node j is less than the capacity of the vehicles. This also implies that if no vehicle travels between the nodes, no passengers can travel between them. Constraints (9)–(11) describe when each node can be serviced. Constraint (12) makes sure that all nodes are serviced within their time window. Constraint (13) guarantees that the ride time is less than the maximum ride time allowed for each request. Constraints (14)–(15) guarantee that the duration time of each vehicle is lower than the maximum duration time. Constraint (16) sets the time window in which the vehicles are allowed to leave the depot. Since the vehicle related variable x_{ij}^k is set binary by constraint (17), the customer related variable y_{ij}^r does not have to be set binary in constraint (18), since the other constraints will force it to a binary value. The fixed route related variable z_{ij} is finally set binary by constraint (19).

As can be seen from the formulation above, the problem is addressed from an operator's perspective with purpose to minimize the total routing cost of the demand responsive vehicles. From a customer's point of view, a journey with one or two transfers is less convenient than a journey without transfer. Since the operator's perspective is taken, it is not considered which alternative solution is the most preferable for the customer. All solutions fulfilling the constraints regarding travel time are considered acceptable for the customer.

What also can be seen from the model is that no synchronization between the dial-a-ride vehicles and time tables of the fixed routes is considered. It is assumed that the fixed routes that are part of the integrated service are so highly frequented that different waiting times depending on when the customer arrives at a transfer location does not have to be considered.

4 Strengthening the mathematical model

The model presented in Sect. 3 grows very rapidly in size with the number of requested journeys. Each request adds $2 + g$ nodes, where g is the number of transfer nodes, to the network and also adds all the associated arcs. In order to solve larger instances of the model, it is very important to be able to reduce the size of the network. Section 4.1 presents some arc elimination rules that reduce the number of arcs, i.e. the number of variables. It is also shown, in Sect. 4.2, how the number of binary variables can be reduced by introducing a new set of variables and in Sect. 4.3 how subtour elimination constraints can be added to the model. Section 4.4 presents additional constraints making it possible to relax the binary restriction on z_{ij}.

4.1 Arc elimination

Some of the arcs in the graph G represent infeasible transportation opportunities. The requirements on time windows and maximum ride times make these arcs impossible to use in a feasible schedule. Therefore, these arcs can be eliminated, which in the proposed formulation directly lead to a reduction of the number of binary variables. First some arc elimination rules previously defined for the DARP are applied, see, e.g., Cordeau (2006), and then some more elimination rules, specific for the IDARP, are presented.

Some arcs connected to depot nodes can always be eliminated. These are the following:

- No arc can go to the node representing the origin depot, i.e., all arcs $(i, 0)$ are infeasible for $i \in N$.
- No arc can go from the node representing the destination depot, i.e., all arcs $(2n + 1, i)$ are infeasible for $i \in N$.
- No arc can go from the origin depot to a drop-off node i, i.e., all arcs $(0, n + i)$ are infeasible for $i \in P$.
- No arc can go from a pick-up node i to the destination depot, i.e., all arcs $(i, 2n + 1)$ are infeasible for $i \in P$.

There are some further elimination rules valid in both DARP and IDARP. These rules refer to arcs between nodes of the same request and arcs that can be eliminated due to time constraints. For some of these elimination rules to be valid, it is assumed that the fixed route service never travels faster than the dial-a-ride service.

- No arc can end at the same node as it starts, i.e., all arcs (i, i) are infeasible for $i \in N$.
- No arc can go from a drop-off node to the pick-up node of the same request, i.e., all arcs $(n + i, i)$ are infeasible for $i \in P$.
- If $e_i + d_i + t_{ij} > l_j$ for $i, j \in N$, the arc (i, j) is infeasible.
- If $t_{ij} + d_j + t_{j,n+i} > \bar{L}_i$ for $i \in P$, $j \in N$, the arcs (i, j) and $(j, n+i)$ are infeasible.
- If the path $\hat{P} = \{j, i, n+j, n+i\}$ is infeasible, so is the arc $(i, n+j)$, with $i, j \in P$.
- If the path $\hat{P} = \{i, n+i, j, n+j\}$ is infeasible, so is the arc $(n+i, j)$, with $i, j \in P$.

In an IDARP there are some more arcs that can be eliminated, all of these are in some way connected to at least one transfer node. As above, it is assumed that the fixed route service never travels faster than the dial-a-ride service. In the rules presented below, S_i is any transfer node belonging to request i.

- In the same way as a pick-up node i cannot be visited after the drop-off node of the same request, a transfer node for the request cannot be visited after the drop-off node. Nor can a pick-up node be visited after a transfer node. All arcs $(n + i, S_i)$ and (S_i, i) are therefore infeasible for $i \in P$.
- If $(i, n + j)$ is infeasible, all arcs (i, S_j) are also infeasible, for $i \in N$, $j \in P$.
- If $i, j \in P$, all arcs (S_i, S_j), $(S_i, n + j)$ and (S_i, j) are infeasible if $(i, n + j)$ is infeasible.
- There is never any reason to why a dial-a-ride vehicle should travel directly between two transfer nodes associated with the same request, i.e. two nodes in C^r. By this reason all arcs (S_i, S_i) are infeasible.

There are also some commonly used elimination rules valid for the DARP that are not valid for the IDARP, simply because it is not certain that the vehicle that picks up a request at the pick-up node also drops the request off at the drop-off node.

4.2 Variable substitution

Since the fleet of vehicles is homogeneous and uses the same depot, and each node can only be visited once, it is not necessary to explicitly know which vehicle visits each node. As long as information is available about which arcs that are used, it is possible to build up all the vehicle itineraries anyway. This makes it possible to reduce the number of binary variables substantially. In order to do this, the following new variables are introduced:

$$v_{ij} = \begin{cases} 1, & \text{if a vehicle travels from node } i \text{ to node } j \\ 0, & \text{otherwise} \end{cases}$$

and related to x_{ij}^k by

$$v_{ij} = \sum_{k \in K} x_{ij}^k \quad i \in N, j \in N, (i, j) \neq (0, n+1) \tag{20}$$

Constraint (20) can be used in the objective function (1) and in constraints (2), (4), (5), (8), (10), and (14) to substitute the x_{ij}^k-variables. The only variables that cannot be substituted are x_{0j}^k since these are needed to set the starting time for each vehicle in constraints (3) and (9). Given the substitutions, constraint (17) can be replaced with

$$\sum_{k \in K} x_{0j}^k = v_{0j} \quad j \in P \cup D \cup C \tag{21}$$

$$v_{ij} \in \{0, 1\} \quad i \in N, j \in N \tag{22}$$

$$x_{0j}^k \in \{0, 1\} \quad j \in N \tag{23}$$

4.3 Subtour elimination

A way to strengthen the model is to introduce constraints that eliminate subtours in the LP-relaxation of the problem. The solution to the LP-relaxation of the problem is used to compute a lower bound on the optimal objective function value of the original problem, and should be as high as possible. In order to increase the objective value of the LP-solution, and thereby increase the lower bound of the original problem, constraints eliminating subtours can be added.

In a feasible solution, only one vehicle can travel between any pair of nodes, and only in one direction. This implies that between any nodes in N, the sum of v_{ij} and v_{ji} must be less than or equal to one, as described in constraint (24).

$$v_{ij} + v_{ji} \leq 1 \quad (i, j) \in A \tag{24}$$

Clusters with at least one node in $P \cup D$ and not including the depot nodes can also be identified. Since at least one vehicle must visit each such cluster in a feasible solution, one can add a constraint saying that each such cluster must have an inflow (or outflow) greater than or equal to one. If $\mathcal{S}$ is the set of all identified clusters and S is one such cluster, constraint (25) describes how subtours corresponding to the clusters can be eliminated.

$$\sum_{i \in S} \sum_{j \in N \setminus S} v_{ij} \geq 1 \quad S \in \mathcal{S} \tag{25}$$

Constraint (25) says that for each identified cluster, the outflow of vehicles from the nodes in the cluster to other nodes must be greater than or equal to one.

The heuristic used to separate the clusters starts with including one pick-up or drop-off node in a cluster S. Then the cluster is expanded by adding a node connected to at least one of the nodes already in the cluster in the LP-solution. After each expansion, the total inflow and outflow of the cluster is calculated and S is added to $\mathcal{S}$ if any of them is less than one. The expansion ends when no more nodes are connected to the cluster. This procedure is repeated for all pick-up and drop-off nodes, and the subtour elimination phase ends when no more clusters are found.

4.4 Transfer node strengthening

The final approach to make the model easier to solve is to introduce two new constraints regarding the transfer nodes. Constraint (26) ensures that at maximum two transfer nodes belonging to a customer can be visited, i.e., no customer is allowed to be transferred between vehicles more than twice. Constraint (27) states that if a demand responsive vehicle arrives at a transfer node, a fixed route either arrives at, or departs from, this node.

$$\sum_{i \in C^r} \sum_{j \in N} v_{ij} \leq 2 \quad r \in R \tag{26}$$

$$\sum_{j \in N} v_{ji} = \sum_{j \in C^r} (z_{ji} + z_{ij}) \quad r \in R, i \in C^r \tag{27}$$

The addition of these constraints also makes it possible to relax the binary restriction on z_{ij}, i.e., constraint (19) can be substituted with constraint (28).

$$z_{ij} \geq 0 \quad r \in R, (i, j) \in (C^r \times C^r) \tag{28}$$

5 An illustrated example

The model described in Sect. 3 has been solved for several small instances. So far, problem instances up to 10 requests have been solved to optimality. It should though be noticed that as stated in Parragh et al. (2008), the number of requests is not a very good indicator of what type of instances can be solved with the model. This is due to the fact that tightly constrained instances are so much easier to solve than instances with less tight constraints.

The model has been implemented in AMPL and solved using CPLEX 11.0.0. The tests were performed on a Windows XP PC with Pentium 4 (3.2 GHz) and 1 Gb of memory. This section will describe one of the test cases, including four requests, three transfer locations and two dial-a-ride vehicles. It also explains how the input and results can be visualized in a GIS.

A GIS can be of great help for creating input files to any kind of routing problem. It can also be a useful tool for displaying the results when the problem is solved. The simplicity in visualizing network-based problems and the handling of such data are two of the advantages of using a GIS for this kind of problem. But the most important advantage is perhaps that most of the input data already is available in this format. Road networks, the location of fixed route services and customer registers are available (at least for all areas in Sweden).

5.1 Input

The GIS used is ArcGIS 9.1 with the network analyst extension. All the nodes in N have been pin placed, with the small exception that for the transfer nodes it is enough to place one node at each geographical location, there is no need of creating all the nodes in C. Each kind of node (pick-up, drop-off, depot and transfer) has been created in its own layer. All interesting and relevant information regarding the nodes, for example time windows, have been stored as attributes of each feature in the layers, thus simplifying the handling of the information.

The input to our test case can be seen in Fig. 1. The test case includes four requests. The pick-up nodes are numbered 1–4 and the drop-off nodes are numbered 5–8. The origin depot has number 0 and the destination depot has number 9. The three transfer locations are numbered 10–12. According to how the set C is defined, there is one node for each request at each geographical transfer location. So for each number representing a transfer location in the figures, there are actually four nodes on top of each other, with the exact same geographical position.

A road network has been used to create a travel cost matrix (available by the use of the network analyst extension) with all relations between all nodes. The calculations of the cost of the different relations have been made directly in the GIS. The travel time of each link in the network has been used as input for these calculations. Consequently, the values in the matrix represent travel times, and are presented in Table 1. Other measures of costs can of course be used, and calculated in the same way.

Since the instance has four requests and three transfer locations (leading to 12 nodes in the set C) there is a total of 22 nodes in set N. These 22 nodes give a total of 484 original arcs. By the arc elimination rules described in Sect. 4.1, this number is reduced to 359. How many arcs that can be eliminated of course depends very much on how the time windows are defined. More narrow time windows and more scattered requests, both in time and space, give a larger number of arcs that can be eliminated. In our test case, the time windows have been defined as in Table 2, where each request has a time window defined for its pick-up node, but none for the drop-off node. For the drop-off nodes, the time windows have simply been calculated as:

$$e_{n+i} = e_i + d_i + t_{i,n+i} \tag{29}$$

$$l_{n+i} = l_i + \bar{L}_i \tag{30}$$

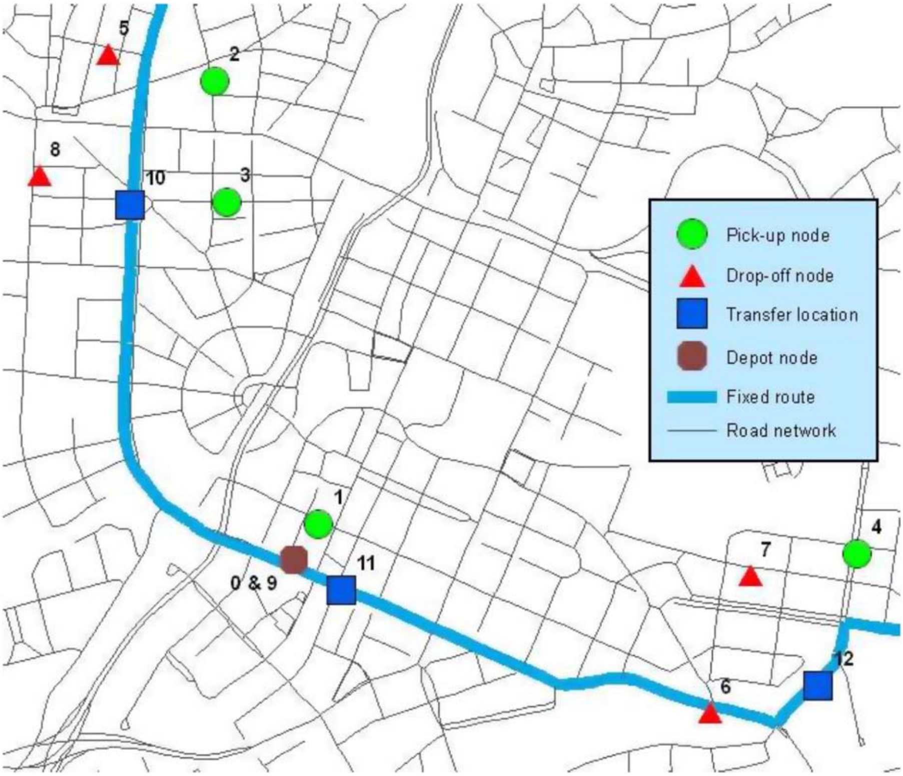

Fig. 1 Input to the test case

Table 1 Travel cost matrix of the test case, created in a GIS

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	23	173	131	195	170	129	160	153	0	120	21	163
1	23	0	166	123	173	179	125	138	168	23	135	26	159
2	173	166	0	57	289	41	268	254	79	173	57	191	301
3	131	123	57	0	258	72	236	222	60	131	29	148	270
4	195	173	289	258	0	318	73	35	306	195	273	174	39
5	170	179	41	72	318	0	296	282	53	170	51	191	329
6	129	125	268	236	73	296	0	48	282	129	248	108	34
7	160	138	254	222	35	282	48	0	271	160	237	139	57
8	153	168	79	60	306	53	282	271	0	153	35	174	316
9	0	23	173	131	195	170	129	160	153	0	120	21	163
10	120	135	57	29	273	51	248	237	35	120	0	141	283
11	21	26	191	148	174	191	108	139	174	21	141	0	142
12	163	159	301	270	39	329	34	57	316	163	283	142	0

Table 2 Time windows for pick-up nodes $i \in P$

i	e_i	l_i
1	50	950
2	350	1250
3	650	1550
4	950	1850

The earliest time a drop-off node can be serviced is described in (29), saying that this is equal to the earliest time the pick-up node of that request could be serviced, plus the service time at that node and the lowest possible travel time to the drop-off node. The latest time a drop-off node can be serviced, as described in (30), is the sum of the latest possible time for servicing the associated pick-up node and the maximum ride time of the associated request.

No earliest and latest times at which vehicles are allowed to leave the depot are used, i.e., e^k and l^k are assumed to be the start and end time of the planning horizon.

5.2 Computational results and solution visualization

To describe the effects of the variable substitution and the subtour elimination constraints presented in Sect. 4.2 and Sect. 4.3, three different tests are made. Test 1 is made without both variable substitution and subtour elimination constraints. In Test 2, the variable substitution is included. Test 3 includes both the variable substitution and the subtour elimination constraints.

Neither Test 1 nor Test 2 did find the optimal solution within the time limit used (7200 seconds). The value of the best solution found in Test 1 is 1058 with a lower bound of 555 which gives a gap of 48%. In Test 2, the value of the best found solution is 1099 with the lower bound 603 giving a gap of 45%. In Test 3, the optimal solution was found after 4165 seconds. The optimal solution value is 880.

The benefits of the subtour elimination is also clearly stated by looking at the LP-relaxation of the different tests. For Test 1 and Test 2, the objective value of the LP-relaxation is 480 and 501 respectively, but for Test 3 it is as high as 828.

The optimal solution to this test case is presented in Table 3. The table also shows which customers that embark or disembark the vehicles at each node. An illustration of the solution is presented in Fig. 2.

As can be seen in Table 3, vehicle one departs from the depot to pick-up request 1, 3 and 2. Thereafter request 1 is dropped off at its destination, and the vehicle proceeds to transfer location 1 (node 10) where request 2 and 3 are transferred to a fixed route service. At the same transfer location, the vehicle then waits for request 4 to arrive with the fixed route service. During this time vehicle two picks up request 2 and 3 at transfer location 2 (node 11) and takes them to their destinations. On this route, request 4 is picked up and taken to transfer location 2, and vehicle two returns to the depot. Vehicle one picks up request 4 at transfer location 1 and proceeds to the requested destination. Thereafter vehicle one also returns to the depot. We can notice that one vehicle gets a quite long waiting time at transfer location 1.

When the optimal routes have been computed they are fed back into the GIS. In the GIS, the exact routes, with the exact links in the road network are visualized.

Table 3 Optimal solution to the test case

Vehicle 1	Node	Node type	Request	Vehicle 2	Node	Node type	Request
	0	depot	–		0	depot	–
	1	pick-up	1		11	transfer	2, 3
	3	pick-up	3		7	drop-off	3
	2	pick-up	2		4	pick-up	4
	5	drop-off	1		6	drop-off	2
	10	transfer	2, 3, 4		11	transfer	4
	8	drop-off	4		9	depot	–
	9	depot	–				

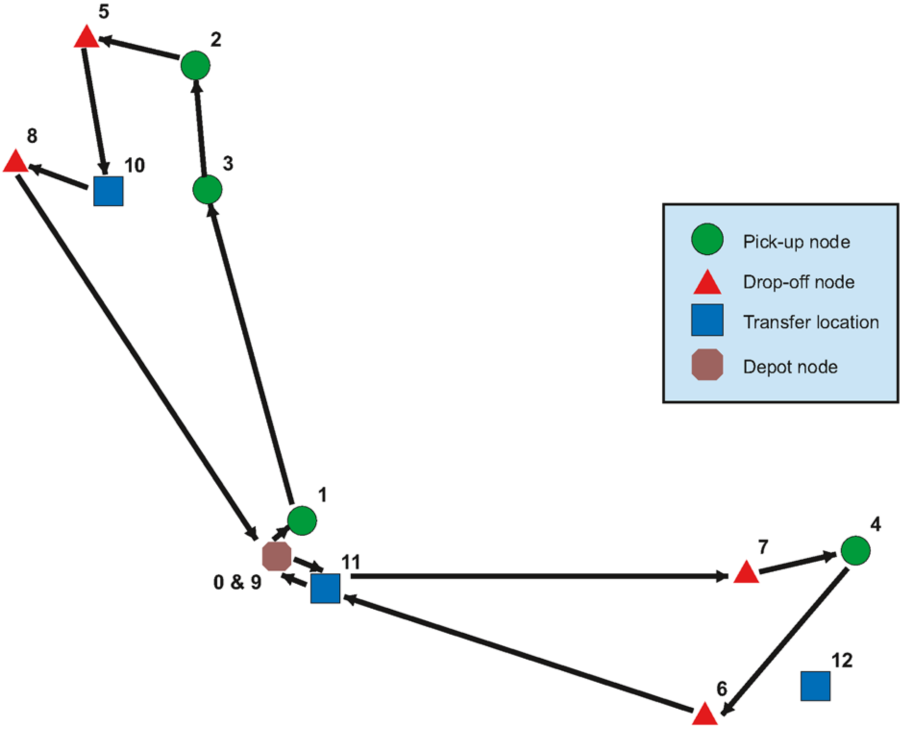

Fig. 2 Visualization of the optimal solution to the test case

This is done by finding the cheapest paths between the nodes in the route, using the network analyst extension. The final results of what these routes look like in the GIS can be seen in Fig. 3.

6 Conclusions

An exact formulation of the Integrated Dial-a-Ride Problem (IDARP) have been introduced. Different ways to reduce the size of the problem have also been presented.

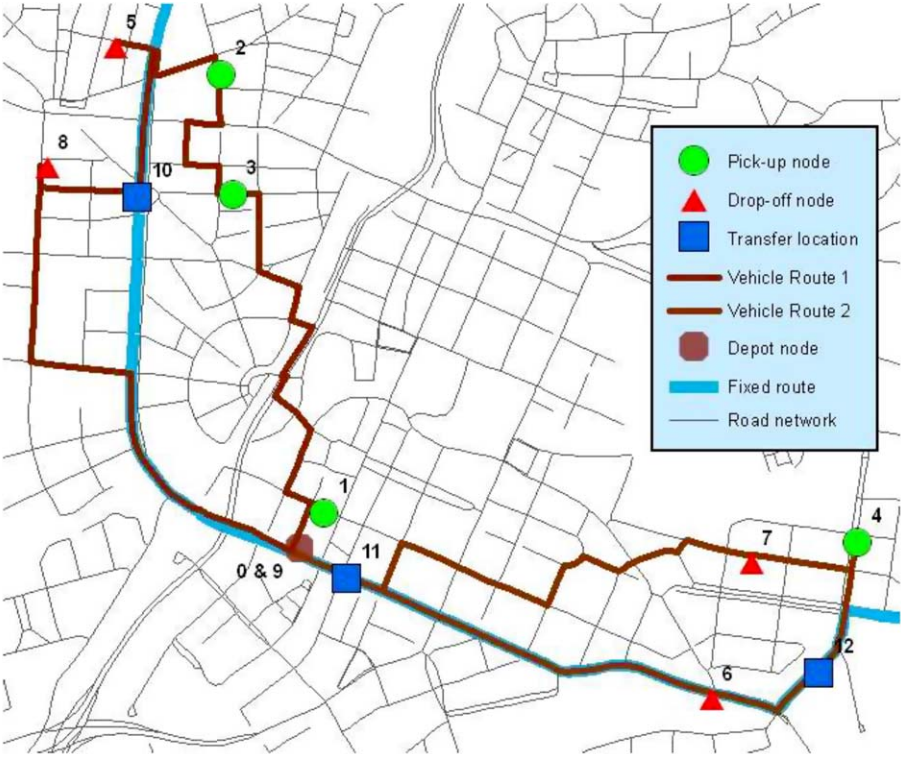

Fig. 3 The final routes of the solution

The model has been tested on several cases and proven to work as intended. Small scale cases can be solved to optimality. In order to solve medium to large sized problem instances, more efficient solution methods need to be developed. The next step in our work of the IDARP is to evaluate how different characteristics of input data, such as time window sizes and dispersion of requests in both time and space affect the complexity of the problem. Another interesting field for future research is to study how different branching techniques affect the computation time. We believe that with further research on suitable branching techniques together with a deeper study of valid inequalities for the IDARP, medium sized problems can be solved. The benefits of using a geographic information system for the data handling and visualization of the results have also been shown.

References

Aldaihani M, Dessouky M (2003) Hybrid scheduling methods for paratransit operations. Comput Ind Eng 45:75–96

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

Baugh J, Kakivaya G, Stone J (1998) Intractability of the dial-a-ride problem and a multiobjective solution using simulated annealing. Eng Optim 30:91–123

Bergvinsdottir K, Larsen J, Jorgensen R (2004) Solving the dial-a-ride problem using genetic algorithms. Technical Report 2004-20, Informatics and Mathematical Modelling, Technical University of Denmark, Denmark

Börjesson M (2001) Erfarenheter av samordning av färdtjänst och sjukresor i Dalsland. Vinnova Rapport 2001:27

Borndörfer R, Grötschel M, Klostermeier F, Küttner C (1997) Telebus Berlin: Vehicle scheduling in a dial-a-ride system. Technical Report SC 97-23, Konrad-Zuse-Zentrum für Informationstechnik, Berlin

Chan S (2004) Metaheuristics for solving the dial-a-ride problem. PhD Thesis, North Carolina State University, USA

Cordeau J (2006) A branch-and-cut algorithm for the dial-a-ride problem. Oper Res 54:573–586

Cordeau J, Laporte G (2003) A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transp Res Part B 37:579–594

Diana M, Dessouky M (2004) A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. Transp Res Part B 38:539–557

Hickman M, Blume K (2001) An investigation of integrated transit service. Research Report SWUTC/01/472840-00023-1, Texas Transportation Institute, USA

Horn M (2004) Procedures for planning multi-leg journeys with fixed-route and demand-responsive passenger transport services. Transp Res Part C 12:33–55

Ioachim I, Desrosiers J, Dumas Y, Solomon M, Villeneuve D (1995) A request clustering algorithm for door-to-door handicapped transportation. Transp Sci 29:63–78

Jaw J, Odoni A, Psaraftis H, Wilson N (1986) A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. Transp Res Part B 20:243–257

Liaw C, White C, Bander J (1996) A decision support system for the bimodal dial-a-ride problem. IEEE Trans Syst Man Cybern Part A 26:552–565

Madsen O, Ravn H, Rygaard JM (1995) A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. Ann Oper Res 60:193–208

Parragh S, Doerner K, Hartl R (2008) A survey on pickup and delivery problems: Part II: Transportation between pickup and delivery locations. J Betriebswirtsch 58:81–117

Psaraftis H (1980) A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. Transp Sci 14:130–154

Psaraftis H (1983) An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. Transp Sci 17:351–357

Ropke S, Cordeau J, Laporte G (2007) Models and branch-and-cut algorithms for pickup and delivery problems with time windows. Networks 49:258–272

Toth P, Vigo D (1997) Heuristic algorithms for the handicapped persons transportation problem. Transp Sci 31:60–71

Uchimura K, Takahashi H, Saitoh T (2002) Demand responsive services in hierarchical public transportation system. IEEE Trans Veh Technol 51:760–766

Wilson N, Sussman J, Wang H, Higonnet B (1971) Scheduling algorithms for dial-a-ride systems. Technical Report USL-TR-71-13, Massachusetts Institute of Technology, USA

Wilson N, Weissberg R, Hauser J (1976) Advanced dial-a-ride algorithms research project: Final report. Technical Report 76-20, Massachusetts Institute of Technology, USA