EXPLORE
IJET International Journal of Engineering & Technology
CENTRE OF PROFESSIONAL RESEARCH PUBLICATIONS
ISSN 2049-3444
International Journal of Engineering and Technology Volume 5 No. 10, October, 2015

Analysis on Model and Algorithm of Static Dial-A-Ride Problems Using Lower Bound

Taehyeong Kim

Senior Researcher, Korea Institute of Civil Engineering and Building Technology, Korea

ABSTRACT

This paper studies a static dial-a-ride problem with time varying travel times, soft time windows, and multiple depots. Kim and Haghani (2011) formulated a static DARP model as a mixed integer programming and in order to validate the model, several random small network problems are solved using commercial optimization package, CPLEX. Three heuristic algorithms based on sequential insertion, parallel insertion, and clustering first-routing second are proposed to solve this problem within a reasonable time for implementation in a real-world situation. It is necessary to find a lower bound of objective function for optimization problems to minimize objective functions. In this paper, a lower bound was found for the developed model by Kim and Haghani (2011). Also, the results of three heuristic methods are compared with the results obtained from exact solution by CPLEX and lower bound to validate and evaluate three heuristic algorithms. Computational results show that three heuristic algorithms are superior compared to the exact algorithm and lower bound solution in terms of the calculation time as the problem size (in terms of the number of demands) increases. Also among the three heuristic algorithms, the heuristic algorithm based on parallel insertion is more efficient than other heuristic algorithms that are based on sequential insertion and clustering first-routing second.

Keywords: *Dial-a-ride problems, Paratransit, Demand responsive system, Lower bound, integer programming*

1. INTRODUCTION

Dial-a-Ride service (also called demand responsive service or paratransit) is the most widely available transit service, with over 7,200 agencies providing transit service in the United States. Dial-a-ride service is comprised of passenger cars, vans or small buses operating in response to calls from passengers or their agents to the transit operator. The operator dispatches a vehicle to pick up the passengers and transport them to their destinations. Most agencies limit this service to disabled persons, their attendants and companions, or seniors.

The Dial-a-Ride Problem (DARP) is NP-hard [1] and belongs to the generic class of vehicle routing and scheduling problems and has been extensively studied over the last 40 years. Several surveys on models and algorithms developed for the DARP can be found in Savelsbergh and Solomon [2], Mitrovic-Minic [3, 4], Desaulniers et al. [5], and Cordeau and Laporte [6, 7]. Generally, all demands are known in advance based on reservations on previous days or subscriptions for regular service in static DARP. Usually this problem needs to be solved before the operations start at the beginning of every day. Before Fu [8, 9], all researchers assumed that travel times in an urban traffic environment are fixed and constant. He proposed improving paratransit scheduling by considering dynamic and stochastic variations in travel time. He found that both dynamic and stochastic variations in travel times had important effects on the quality of the schedules, and an appropriate consideration of these variations in the scheduling process could substantially improve the reliability and productivity of the schedules.

Kim and Haghani [10] formulated a static DARP model as a mixed integer programming and in order to validate the model, several random small network problems are solved using commercial optimization package, CPLEX. Three heuristic algorithms based on sequential insertion, parallel insertion, and clustering first-routing second are proposed to solve this problem within a reasonable time for implementation in a real-world situation. It is necessary to find a lower bound of objective function for optimization problems to minimize objective functions. In this paper, a lower bound was found for the developed model by Kim and Haghani [10]. Also, the results of three heuristic methods are compared with the results obtained from exact solution by CPLEX and lower bound to validate and evaluate three heuristic algorithms.

2. MODEL AND ALGORITHM OF STATIC DARP

In this paper, accommodating only one type of demands which are known in advance is considered. Also, time varying travel times are considered because link flow speeds are not fixed during the service period and fluctuate. Like a real-world situations, more than one depot, from where vehicles of a fleet can start operating, are considered.

2.1 Model

As noted earlier, this paper is focused on static dial-a-ride problem with time varying travel times, multi-depots, and heterogeneous vehicles. The detailed explanation about the model and algorithm for this problem can be found in Kim and Haghani [10].

2.1.1 Assumption and Limitations

All demands are known in advance because of customers' reservations before the trip day. Every demand has a demand request time (a desired pick-up time and drop-off time), pick-up node, drop-off node, and load (regular passengers, wheelchair passengers, and transferrable wheelchair passengers).

Travel times are subject to change according to the time of the day. It is assumed that we have link flow speeds within each time interval (10 minutes) which is based on historical data in network. Given link flow speeds, it is possible to calculate the expected travel time between origin and destination at starting time using a time dependent shortest path algorithm.

The routing plan is developed based on the demands which are known in advance. It is assumed that there is the maximum route duration which cannot be exceeded by any vehicle. Also, the ride time for a customer who is picked up at pick-up node cannot exceed his or her maximum ride time.

It is assumed that there are multi depots. The number of available vehicles at a depot and locations of depots are known. Each vehicle has its own capacity, and the vehicles are not homogeneous. Three different types of handicapped patients are considered in this research as follows: the ambulatories who use regular seats, those who use wheelchairs, and those who use transferable wheelchairs.

Using soft time windows allows the vehicles to arrive at the pick-up and drop-off nodes before or after the time interval that is designated for service. If there is any customer on board, it is not allowed to wait for servicing customers at the node.

2.1.2 Objective Function and constraints

The objective of this problem is to minimize the total cost composed of the service provider's cost and the customers' inconvenience cost. The service provider's cost includes the fixed costs for the used vehicles, the routing costs, and vehicle waiting cost. The user inconvenience cost includes customers' excess ride time cost and delayed service cost.
The constraints in this model can be divided into six groups: Depot, capacity, precedence and coupling, routing, and time window, and no-waiting on board constraints.

2.2 Heuristic Algorithms

In this paper, the approach for solving this model can be divided into two phases including a construction phase and an improvement phase. In the first phase, feasible routes are constructed and in second phase the routes are improved.

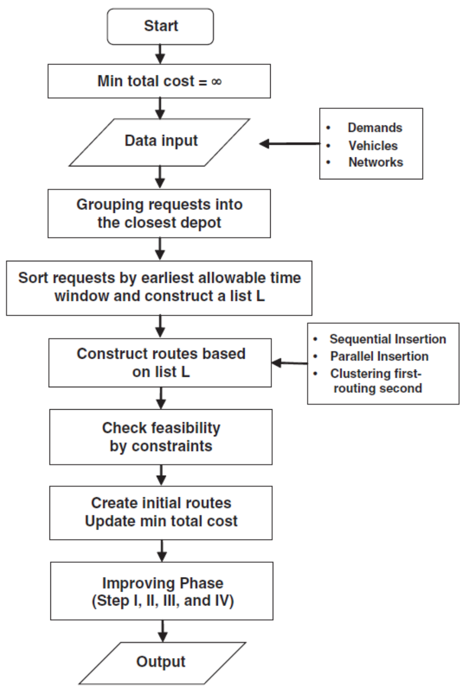

Fig 1: The framework of the heuristic for static DARP [10]

2.2.1 Construction

Before construction phase, all demands are grouped into the closest depot and sorted by earliest allowable time window.
In construction phase, we build a set of feasible routes for each depot starting from the information that define dial-a-ride problem by three different heuristic methods such as sequential insertion (HSI), parallel insertion (HPI), and clustering first-routing second (HCR) [10].

2.2.2 Improvement

After obtaining a feasible solution in the construction phase, the solution is improved through 4steps of improvement phase.

(1) Step I: Acceptable waiting and delay time
While the violation of time window is allowed in this model, the service quality by the maximum acceptable waiting and delay time in this phase can be adjusted. After constructing phase, in initial solution, there may be bad routes in which some customers have unreasonable waiting or delay time. In this phase, all routes are improved to satisfy maximum acceptable waiting and delay time.

(2) Step II: Remove one insert one
For the local improvement procedure to explore the local region, a trip insertion operator is applied at this phase. Trip insertion removes a given trip from a route and then inserts it into the best position of another route. Improving phase II is repeated up to two times because in most cases it is observed that after two times of repetition the objective functions converge.

(3) Step III: Combining vehicles
After improving step II, there may be some routes which have a few demands. These routes can be combined into other routes that can accept customers from them without violating constraints.

It is allowed for a vehicle to be combined into another vehicle that starts from a different depot.

(4) Step IV: Adjusting vehicle starting times
At this step, the waiting time at a demand node in each route is checked. If there is any waiting time at a demand node in a route, the starting time of the vehicle serving that route is adjusted using marginal time. The marginal time at a node is defined as the maximum delay in arrival at the current node that does not cause violation of the time windows at the following nodes.

3. LOWER BOUND

In this section, the approach to find a lower bound is presented. The original formulation is reformulated with new variables and constraint using LP relaxation. Then, the new mixed integer programming problem is solved. For finding the lower bound the method used by Jung [11] for pickup and delivery problem is modified for DARP.
In the lower bound solution procedure, it is tried to solve larger problems, although the results are not the exact solutions.

3.1 Procedure

The strategy of the lower bound solution procedure is to find a way that minimizes the number of integer variables. The simplest way to minimize the number of integer variable is LP relaxation. It is necessary that new variables and constraints are added to relaxed formulation in order to provide a good lower bound since the objective function of relaxed formulation is too low compared to the known optimal value for very small problems when the original formulation is relaxed without any changes and the problem is solved.

In the original formulation, there are two kinds of binary variables. These are x_{ij}^{kt} and y_{ik}. For relaxation, x_{ij}^{kt} and y_{ik} are changed to general integer variables. And, new variables S_{ij}^{kt} , Z_i^t , and V_i^k are added as follows:

S_{ij}^{kt} = 1 if vehicle *k* starts from depot *i* to demand *j* at time *t*,
= 0 otherwise
Z_i^t = 1 if vehicle *k* departs from node *i* at time *t*,
= 0 otherwise
V_i^k = 1 if demand *i* is serviced by vehicle *k*,
= 0 otherwise

New constraints (1) and (2) are added. Constraints (1) states that new variable S_{ij}^{kt} is equivalent to x_{ij}^{kt} when *i* belongs to the set of starting depot.

$$S_{ij}^{kt} = x_{ij}^{kt} \qquad i \in S, j \in \Phi \qquad (1)$$
$$\sum_{i\epsilon S}\sum_{j\epsilon\Phi}\sum_{t\epsilon\alpha}^{\omega}\sum_{k\epsilon V} S_{ij}^{kt} \geq 1 \qquad (2)$$

The constraint (7') in the original formulation [10] is replaced as expression (3).

$$\sum_{i\epsilon S}\sum_{j\epsilon P}\sum_{t=\alpha}^{w} x_{ij}^{kt} \leq 1 \qquad k \in V \qquad (7')$$
$$\sum_{i\epsilon S}\sum_{j\epsilon P}\sum_{t=\alpha}^{w} S_{ij}^{kt} \leq 1 \qquad k \in V \qquad (3)$$

Also new constraint (4), (5), and (6) are added.

$$V_i^k = \sum_{j\in N_s}\sum_{t=\alpha}^{w} x_{ij}^{kt} \qquad i \in \Phi, k \in V \qquad (4)$$

Constraint (5) states that the new variable Z_i^t is the sum of all connection from demand node *i* to any demand node at time *t*.

$$Z_i^t = \sum_{k\in V}\sum_{j\in N_s} x_{ij}^{kt} \qquad i \in \phi,\ t \in T \qquad (5)$$
$$\sum_{t=\alpha}^{\omega} Z_i^t = 1 \qquad i \in \phi \qquad (6)$$

Also constraint (27') for waiting penalty and constraint (29') for delay penalty in the original formulation can be rewritten as expression (7) and (8) using Z_i^t.

$$w_j = Max\left(0, a_j - \sum_{k\in V}\sum_{i\in\eta}\sum_{t=\alpha}^{\omega} x_{ij}^{kt}\,(t + R_{ij}^t)\right) \quad j \in \phi \quad (27')$$
$$d_j = Max\left(0, \sum_{k\in V}\sum_{i\in\eta}\sum_{t=\alpha}^{\omega} x_{ij}^{kt}\,(t + R_{ij}^t) - b_j\right) \quad j \in \phi \quad (29')$$
$$w_i = Max(0, a_i - \sum_{t=\alpha}^{\omega} tZ_i^t - s_i) \quad j \in \phi \quad (7)$$
$$d_i = Max(0, \sum_{t=\alpha}^{\omega} tZ_i^t - s_i - b_i) \quad j \in \phi \quad (8)$$

The fixed cost part of the objective function in the original formulation is modified using S_{ij}^{kt}. The underlined part in expression (9) shows the modified part as follows:

$$Min \quad f_c \times \underline{\sum_{k\in V}\sum_{i\in S}\sum_{j\in P}\sum_{t=\alpha}^{\omega} S_{ij}^{kt}} + R_c \times \sum_{i\in\tau}\sum_{j\in N_s}\sum_{k\in V}\sum_{t=\alpha}^{\omega}\left(R_{ij}^t \times x_{ij}^{kt}\right) + P_w \times \sum_{i\in\phi} w_i$$
$$+ p_e \times \sum_{i\in P}\left(\sum_{k\in V}\sum_{l\in N_s}\sum_{t=\alpha}^{\omega}(t - s_{n+i})x_{n+i,l}^{kt} - \sum_{k\in V}\sum_{j\in\phi}\sum_{t=\alpha}^{\omega} t x_{ij}^{kt} - \sum_{t=\alpha}^{\omega}\left(R_{i,n+i}^t \times \sum_{k\in V}\sum_{j\in\phi} x_{ij}^{kt}\right)\right)$$
$$+ P_d \times \sum_{i\in\phi} d_i \qquad (9)$$

4. COMPUTATIONAL STUDY

In this section, the proposed model is validated through solving a set of small test problems [10] by an exact method using a commercial package, CPLEX. Also, the results of the exact method and lower bound solutions are compared with the results of the heuristic algorithms that are developed for the model in the research [10]. Also the performances of three heuristic algorithms are analyzed in this section.
The heuristic algorithms were coded in C++. All computations were carried out on a machine with 2.0GHZ Intel Core 2 Duo CPU and 3GB memory in Windows XP environment.

4.1 Test problems

4.1.1 The characteristic of problem instances

The exact method can solve problems with a few customers that have to be serviced with a few vehicles. We assume that the service area is 20 miles by 20 miles and there are two depots. The location of depot 1 is (7, 10) and depot 2 is (13, 10). The demands are generated at random over the service area. There are 3, 4, and 5 customers with 10 and 15 time intervals respectively. Each combination of number of demand nodes and the time intervals has three cases of examples. Interval length is 6 minutes. For the 10 time interval case, time period is from 9 am to 10am. For the 15 time interval case, time period is from 9 am to 10:30 am.

4.1.2 Parameter settings

The duration of time window is 12 minutes, maximum route duration for 10 time intervals is 60 minutes and for 15 time interval is 90 minutes, maximum acceptable waiting and delay time is 30 minutes, the fixed cost for used vehicle is \$10,000/vehicle, the travel cost is \$1/minute, the penalty cost for waiting time is \$0.5/minutes, the penalty cost for delay time is \$0.5/minute, and the penalty cost for customers' excess ride time is \$0.5/minute.

4.2 Computational results

In this section, the results from exact solution method, lower bound solution method, and three heuristic methods are presented. The gaps between the exact solution (E), lower bound solution (LB), and three heuristics solutions are calculated as follows:

Total cost gap between HSI and exact solution = (HSI – E)/E*100
Total cost gap between HPI and exact solution = (HPI – E)/E*100
Total cost gap between HCR and exact solution = (HCR – E)/E*100
Total cost gap between HSI and LB solution = (HSI – LB)/LB*100
Total cost gap between HPI and LB solution = (HPI – LB)/LB*100
Total cost gap between HCR and LB solution = (HCR – LB)/LB*100
Total cost gap between LB and exact solution = (E – LB)/LB*100
Calculation time ratio between HSI and exact solution =E/HSI
Calculation time ratio between HPI and exact solution =E/HPI
Calculation time ratio between HCR and exact solution =E/HCR
Calculation time ratio between HSI and LB solution =LB/HSI
Calculation time ratio between HPI and LB solution =LB/HPI
Calculation time ratio between HCR and LB solution =LB/HCR
Calculation time ratio between LB and exact solution =E/LB

Each combination of the number of customers and the time intervals has three cases of examples. In results, the average value of the three examples for each combination is calculated. Tables 1 and 2 show the comparison of the calculation times for the exact method, the lower bound, and the three heuristic algorithms. As the number of customers exceeds 3 with service period of 10 time intervals, the calculation time of exact method increases exponentially and becomes unreasonable.

When the original problem without integer relaxation is solved, the largest problems that can be solved within a reasonable computational time are problems with 5 demands and 10 time intervals. It means that the largest DARP problem size that could be solved in a reasonable time by exact method was 5 customers with service period of 10 time intervals.

For most of the cases, the three heuristic algorithms solved the problems within less than 0.2 second while the exact method could not solve the problem which has 5 customers and 15 time intervals. For example, in case of 5customers and 10 time intervals, HPI solved the problem within 0.12 seconds, HSI solved the problem within 0.16 seconds and HCR solved the problem within 0.17 seconds while the exact method solved the problem within about 139 minutes.

To get the exact solution, we spent about 5091 times as much time as required for the HCR solution for the 5 customers and 10 time intervals. For small problems, the three heuristics algorithms solved the test problems faster than the lower bound and the exact method with almost the same objective function values. Figure 2 shows the comparison of HPI and the lower bound solutions.

Table 1: The comparison of results of calculation times (I)

Number of Customers	Number of Time interval	Calculation Times (seconds)				
		E	LB	HCR	HSI	HPI
3	10	8.08	6.31	0.16	0.15	0.14
3	15	62.12	9.94	0.16	0.14	0.14
4	10	56.30	63.30	0.17	0.17	0.15
4	15	1410.43	138.37	0.12	0.12	0.11
5	10	867.18	3563.34	0.17	0.16	0.12
5	15	-	4717.96	0.14	0.13	0.12

E: Exact solution
LB: Lower bound solution
HCR: Heuristic algorithm based on Clustering first-Routing second
HSI: Heuristic algorithm based on Sequential Insertion
HPI: Heuristic algorithm based on Parallel Insertion

Table 2: The comparison of results for calculation times (II)

Number of Customers	Number of Time interval	Calculation time ratio			
		E& LB	HCR& E	HSI& E	HPI& E
3	10	1.28	50.74	53.35	56.43
3	15	6.25	426.6	473.42	500.66
4	10	0.89	330.27	340.18	372.77
4	15	10.19	11328.77	11574.97	12991.39
5	10	0.24	5091.08	5410.86	7115.81
5	15	-	-	-	-

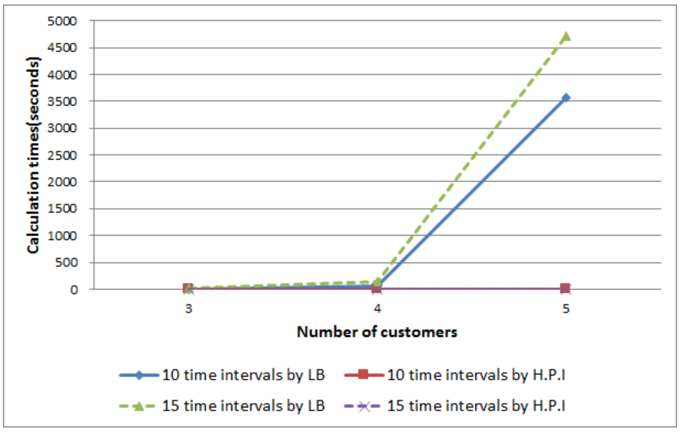

Fig 2: The comparison of the LB and HPI heuristic

Tables 3, 4, and 5 show the comparison of the objective function values for the exact method, the lower bound, and the three heuristic algorithms. The gaps of the objective function value between the exact method and the three heuristic algorithms are less than 0.006%. For example, the gap range of the objective function value between the exact method and HSI is 0.001% to 0.006% and the gap range of the objective function value between the exact method and HPI is 0.001% to 0.003%.

The gaps of the objective function value between the lower bound and the three heuristic algorithms are less than 0.014%. For example, the gap range of the objective function value between the lower bound and HSI is 0.007% to 0.014% and the gap range of the objective function value between the lower bound and HPI is 0.007% to 0.014%. The gap range of the objective functions value between the exact solution and the lower bound is 0.006% to 0.013%.

Table 3: The comparison of results for objective functions (I)

Number of Customers	Number of Time interval	Objective Functions				
		E	LB	HCR	HSI	HPI
3	10	30015.3	30013.0	30015.7	30017.0	30015.7
3	15	30019.2	30016.2	30020.0	30019.7	30019.7
4	10	40023.3	40018.2	40023.7	40023.7	40023.7
4	15	36689.0	36686.3	36690.7	36690.0	36690.0
5	10	50028.3	50025.3	50028.8	50028.8	50028.8
5	15	-	40024.2	40027.2	40027.5	40027.2

Table 4: The comparison of results for objective functions (II)

Number of Customers	Number of Time interval	Objective Functions			
		E& LB	HCR& E	HSI& E	HPI& E
3	10	0.008	0.001	0.006	0.001
3	15	0.010	0.003	0.002	0.002
4	10	0.013	0.001	0.001	0.001
4	15	0.007	0.005	0.003	0.003
5	10	0.006	0.001	0.001	0.001
5	15	-	-	-	-

Table 5: The comparison of results for objective functions (III)

Number of Customers	Number of Time interval	Objective Functions		
		HCR&LB	HSI&LB	HPI&LB
3	10	0.008	0.006	0.001
3	15	0.010	0.002	0.002
4	10	0.013	0.001	0.001
4	15	0.007	0.003	0.003
5	10	0.006	0.001	0.001
5	15	-	-	-

Among the three heuristic algorithms, the heuristic algorithm based on parallel insertion has the best performance based on calculation times and objective function values. For objective function values, HPI, HSI and HCR almost have the same objective function values within less than 0.004% of gap for all cases and HPI has a little better objective function value than HSI and HCR. For calculation times, there are subtle differences in the three heuristic algorithms and HPI solved the problems a little faster than HSI and HCR. It can be said that the proposed heuristic algorithms work well in static DARP. They produce good results within a reasonable calculation time.

4. CONCLUSIONS

In this paper, a lower bound was found for the developed model by Kim and Haghani [10]. Also, the results of three heuristic methods are compared with the results obtained from exact solution by CPLEX and lower bound solution to validate and evaluate three heuristic algorithms. Computational results show that three heuristic algorithms are superior compared to the exact algorithm and lower bound solutions in terms of the calculation time as the problem size (in terms of the number of demands) increases. Also among the three heuristic algorithms, the heuristic algorithm based on parallel insertion is more efficient than other heuristic algorithms that are based on sequential insertion and clustering first-routing second.

REFERENCES

[1] J. Baugh, G. Kakivaya, and J. R. Stone (1998), Intractability of the dial-a-ride problem and a multiobjective solution using simulated annealing, *Engineering Optimization*, Vol. 30, pp. 91-123.

[2] M.W.P. Savelsbergh and M. Solomon (1995), The general pickup and delivery problem, *Transportation Science*, Vol. 29, No. 1, pp. 17-29.

[3] S. Mitrovic-Minic (1998), Pickup and delivery problem with time window: A Survey, SFU CMPT TR 1998-12, Simon Fraser University, Canada.

[4] S. Mitrovic-Minic (2001), The dynamic pickup and delivery problem with time window, Ph. D. Dissertation, Department of Computing Science, Simon Fraser University, Canada.

[5] G. Desaulniers, J. Desrosiers, A. Erdmann, M.M. Solomon, and F. Soumis (2002), VRP with pickup and delivery, in Toth, P. and Vigo, D. (eds.) The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, pp. 225-242.

[6] J.-F. Cordeau and G. Laporte (2003), The dial-a-ride problem: Variants, modeling issues and algorithms, *4OR: A Quarterly Journal of Operations Research*, Vol. 1, No. 2, 2003, pp. 89-101.

[7] J.-F. Cordeau and G. Laporte (2007), The dial-a-ride problem: models and algorithms, *Annals of Operations Research*, Vol. 153, No. 1, 2007, pp. 29-46.

[8] Fu, L. (1999), Improving paratransit scheduling by accounting for dynamic and stochastic variations in travel time, *Transportation Research Record,* No. 1666, pp. 74-81.

[9] Fu, L. (2002), Scheduling dial-a-ride paratransit under

time-varying, stochastic congestion. *Transportation Research Part B*, Vol. 36, pp. 485-506.

[10] T. Kim and A. Haghani (2011), Model and algorithm considering time-varying travel times to solve static multidepot dial-a-ride problem, *Transportation Research Record: Journal of the Transportation Research Board*, No. 2218, pp. 68-77.

[11] S. Jung (2000), A genetic algorithm for the vehicle routing problem with time-dependent travel times, Ph.D. dissertation, University of Maryland.

