Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

A Robust Solution Approach to the Dynamic Vehicle Scheduling Problem

Dennis Huisman, Richard Freling, Albert P. M. Wagelmans,

To cite this article:
Dennis Huisman, Richard Freling, Albert P. M. Wagelmans, (2004) A Robust Solution Approach to the Dynamic Vehicle Scheduling Problem. Transportation Science 38(4):447-458. https://doi.org/10.1287/trsc.1030.0069

TRANSPORTATION SCIENCE
Vol. 38, No. 4, November 2004, pp. 447–458
ISSN 0041-1655 | EISSN 1526-5447 | 04 | 3804 | 0447

DOI 10.1287/trsc.1030.0069

A Robust Solution Approach to the Dynamic Vehicle Scheduling Problem

Dennis Huisman, Richard Freling,* Albert P. M. Wagelmans
Erasmus Center for Optimization in Public Transport (ECOPT), and Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, NL-3000 DR Rotterdam, The Netherlands {huisman@few.eur.nl, wagelmans@few.eur.nl}

This paper presents a solution approach to the dynamic vehicle scheduling problem. This approach consists of solving a sequence of optimization problems, where we take into account different scenarios for future travel times. We discuss the potential benefit of our approach compared to the traditional one, where the vehicle scheduling problem is solved only once for a whole period and the travel times are assumed to be fixed. Because in the multiple-depot case we cannot solve the problem exactly within reasonable computation time, we use a "cluster-reschedule" heuristic where we first assign trips to depots by solving the static problem and then solve dynamic single-depot problems. We use new mathematical formulations of these problems that allow fast solution by standard optimization software. Results of a computational study with real-life data are presented, in which we compare different variants of our approach and perform a sensitivity analysis with respect to deviations of the actual travel times from estimated ones.

Key words: vehicle scheduling; dynamic scheduling; public transport; stochastic travel times
History: Received: June 2001; revision received: April 2002; accepted: August 2002.

Introduction

In this paper, we consider a new way of looking at the vehicle scheduling problem, one of the main scheduling problems of a public transport company. Figure 1 shows the relation between the four operational planning problems in the traditional planning process at a public transport company.

The input consists of decisions about which routes or lines to operate and how frequently. Also known are the travel times between various points on the route. Based on the lines and frequencies, timetables are determined resulting in *trips* with corresponding start and end locations and times. The second planning process is the vehicle scheduling problem, which we will define in the next section. As a result of solving this problem, we obtain a number of vehicle blocks that essentially correspond to assignments of trips to vehicles. On each vehicle block a sequence of tasks can be defined, where each task needs to be assigned to a working period for one crew (a crew duty) in the crew-scheduling process. Rosters are constructed from crew duties during the crew-rostering process. Traditionally, this process is done once for every timetable period, but in reality travel times are not fixed, which means that the vehicle and crew schedules cannot be executed exactly. This may result in trips starting late.

In recent years, it has become much more important for public transport companies to provide an adequate service level to their customers. This is due to privatization and the growing competition in the public transport market. For instance, in the Netherlands, public transport companies (will) sign a contract with the government to provide transport in a certain area that is valid for a limited period. The contract specifies minimum service levels. If these are not met, a fee is due and the contract may not be renewed. For example, this can be the case if there are too many delays. So it is very important for public transport companies to build robust schedules that limit the number of possible delays.

Connexxion, the largest bus company in the Netherlands, provides services for suburban and interregional transport, especially in highly populated areas with a lot of traffic jams. The company experiences a significant number of trips starting late. Therefore, it is studying the possibility of using a dynamic planning process for vehicle and crew scheduling. This has motivated us to develop algorithms to support these processes.

In this paper we only focus on dynamic vehicle scheduling. There are three measurements that we consider throughout the paper, namely the number of vehicles used, the percentage of trips starting late, and a "virtual" measure for delay costs. For the models we will specify objective functions containing these different measurements. The paper is organized as

*In memoriam: Richard Freling passed away on January 29, 2002, at the age of 34.

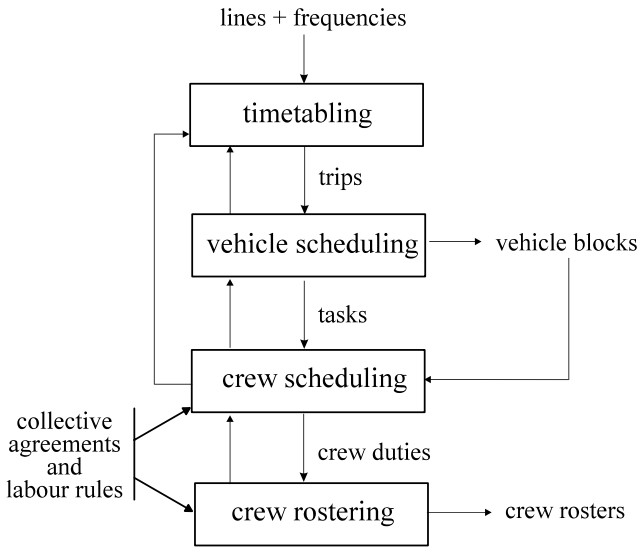

Figure 1 Traditional Planning Process

follows. Section 1 deals with the problem definition and discusses the potential benefit of using dynamic vehicle scheduling. In §2 we present a nonstandard formulation of the static vehicle scheduling problem. This formulation will reappear in an extended form in §3, where we discuss our approach to dynamic vehicle scheduling. We present results of computations with real-life data from Connexxion in §4. In §5 we state our conclusions.

1. Vehicle Scheduling Problem

In this section, we discuss the vehicle scheduling problem (VSP) and the potential benefit of solving a dynamic problem. Furthermore, we give a brief literature review about dynamic approaches in general.

1.1. Problem Definition

In the (multiple-depot) (MD)VSP, the total vehicle costs have to be minimized subject to the following constraints.

- Every trip has to be assigned to exactly one vehicle.
- Some trips have to be assigned to vehicles from a certain set of depots.
- Every vehicle is associated with a single depot.

In some cases there are also other constraints, such as depot capacity constraints, which specify for every depot a maximum number of vehicles. We do not consider these type of constraints in this paper. However, they can easily be incorporated.

The vehicle costs consist of a fixed component for every vehicle and variable costs for idle and travel time. Vehicles may return to their own depots between two trips if there is enough time to do this.

A deadhead is a period that a vehicle is moving to or from the depot, or a period between two trips where a vehicle is outside of the depot (possibly moving without passengers).

Table 1 Data of Trips

Trip	Start time	End time	Start location	End location
1	9:00	10:00	B	A
2	9:15	10:00	C	A
3	10:05	11:05	A	B
4	10:15	11:00	A	C

1.2. Potential Benefit of Dynamic Vehicle Scheduling

Traditionally, the VSP is solved a few months before the new timetable starts, and it will not be changed for the whole period that the timetable is valid. The disadvantage of this approach is that when the schedules are executed and there is a delay at a certain moment, the trip following a delayed trip may start late. Of course, one may try to guard against this problem by adding a fixed buffer time to the travel times, but this buffer will also be present on days that it is not needed, which may cause inefficiencies. The following simple example shows that it may not be obvious how to choose the buffer times.

Example 1. We have four trips—1, 2, 3, and 4—and three locations, A, B, and C. Table 1 gives the start time, end time, start location, and end location of these trips. We minimize the number of vehicles, while we have only one depot.

There are two static optimal solutions. Suppose we choose the solution where Trips 1 and 3 are assigned to Vehicle 1, and Vehicle 2 does Trips 2 and 4.

Furthermore, suppose that Trip 1 has a delay at arrival of 10 minutes, which means that it arrives at 10:10. Then Trip 3 would start five minutes late if we use the schedule above. Even if we had added a buffer time of five minutes (or less) to every trip, the static optimal solution would not change, which means that Trip 3 would still start late. If we had introduced a buffer time of six minutes (or more), we would need three vehicles to complete these trips. Also note that there is another optimal solution for this example, where two vehicles are used and there are no trips starting late: Vehicle 1 does Trips 1 and 4, and Vehicle 2 does Trips 2 and 3.

If the VSP is solved dynamically, which means that we reschedule a few times per day, that is, we reassign vehicles to trips, we may be able to prevent the delays at the start of a trip in many cases. In the above example we could reschedule as soon as we know Trip 1 has a delay and Trip 2 arrives on time, such that Trip 3 can start at the right time. However, there are also some disadvantages to this approach. For example, the schedules are known just before they should be executed, which requires a different way of working and thinking from the company's planners and

managers. Furthermore, a fast and reliable communication and information system is needed between the vehicles and the planners. Of course, in reality a vehicle also needs a driver, which we do not consider in this paper.

1.3. Literature Review

To the best of our knowledge, there is no literature about the dynamic VSP. Therefore, we only discuss here briefly some literature on solving other optimization problems using dynamic models and solution approaches for these models. For a general survey about dynamic and stochastic models, we refer to Powell et al. (1995), who explain why it may be useful to use dynamic models instead of static ones for many problems in the field of transportation and logistics. Furthermore, these authors discuss a lot of different methods to deal with uncertainty.

Powell et al. (2000) explain that because of randomness in travel times, optimal solutions of a sequence of deterministic crew scheduling and vehicle-routing problems may in reality lead to overall nonoptimal solutions. They argue that it is better to use algorithms that are more local in nature, for example, greedy heuristics.

The literature about dynamic optimization problems can be divided in two parts. One part deals with online approaches, as introduced by Sleator and Tarjan (1985). This is especially used in the field of machine-scheduling problems. In the last decade, also in the vehicle-routing area, this type of approaches got more attention. For instance, Ichoua et al. (2000) discuss a tabu search heuristic for real-time vehicle dispatching. In the field of passenger transport, this type of approach is used in dial-and-ride problems (see, e.g., Madsen et al. 1995).

The other part deals with explicit stochastic models. Most often, stochastic programming is used to tackle uncertainty (see, e.g., Laporte and Louveaux 1998 for an application in the area of vehicle routing). Recently, Yen and Birge (2002) have used stochastic programming to solve airline crew-scheduling problems to get more robust schedules.

Online approaches can also be used to solve dynamic VSPs, but we have a lot of information about the future, which we may use in the approach. The start time, start location, and end location of all trips are known. We even know the end times of the trips to some extent. Therefore, our solution approach is based more on the ideas of stochastic programming than on the ones of online algorithms.

2. Static Vehicle Scheduling

In this section, we discuss the static VSP (S-VSP) and present a nonstandard mixed-integer programming formulation of this problem. This formulation will reappear in an extended form in the next section, where we discuss dynamic vehicle scheduling.

For the single-depot case (S-SDVSP), we can use algorithms described in Freling et al. (2001) or Löbel (1997). The multiple-depot case (S-MDVSP), has been shown to be NP-hard by Bertossi et al. (1987). There is a lot of literature on the multiple-depot case. Some models are based on set partitioning formulations (see, e.g., Ribeiro and Soumis 1994), but most of them are based on multicommodity flow formulations (see, e.g., Löbel 1997, Mesquita and Paixão 1999). We will, however, give a formulation that is similar to the formulation proposed by Haase et al. (2001) in the context of integrated vehicle and crew scheduling.

Let D and $N = \{1, 2, \ldots, n\}$ denote the set of depots and trips, respectively. Let N^d be the subset of all trips that are allowed to be assigned to depot $d \in D$, and define D_i as the subset of depots to which trip $i \in N$ can be assigned. Let r_d and t_d denote the source and sink, respectively, of the network corresponding to depot d. This network is denoted by $G^d = (N^d \cup r_d \cup t_d, A^d)$, where A^d is the set of arcs between two compatible trips in N^d, from r_d to every trip in N^d and from every trip in N^d to t_d.

Figure 2 shows an example of such a vehicle network with two depots and seven trips, where $N^1 = \{1, 2, 4, 5, 6\}$ and $N^2 = \{2, 3, 5, 6, 7\}$. Notice that, for reasons of clarity, not all arcs have been drawn.

Let c_{ij}^d be the vehicle cost of arc $(i, j) \in A^d$, which is usually some function of travel and idle time, and let c be the fixed vehicle cost.

We could now give a typical multicommodity flow formulation of the S-MDVSP. We will, however, present a different formulation that, in general, requires fewer variables at the expense of a relatively small number of additional constraints. This formulation, which is valid under a certain assumption, turns out to be easier to solve when standard mixed-integer programming software is used. Slightly different formulations can be found in Lamatsch (1992) and Desrosiers et al. (1995).

The idea is to reduce the size of the arc sets, which may be very large. First, we assume that a vehicle always returns to the depot between two consecutive trips if this is possible and not more expensive. In that

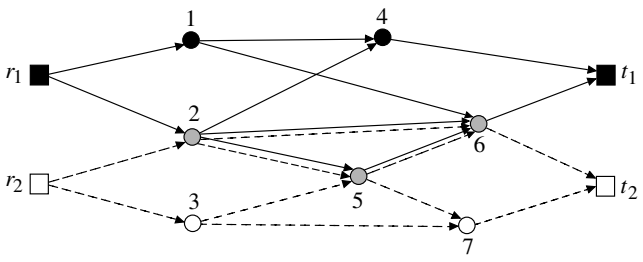

Figure 2 Example—Network G^1 and G^2

case, the arc between the trips is called a long arc; the other arcs between trips are called short arcs. If we further assume that there are no costs involved when a vehicle is idle at the depot, we can delete all long arcs. Let A^{d*}, $d \in D$, denote the part of A^d without long arcs.

Let tp^{dh}, $h = 1, 2, \ldots, m$ be the time points at which a vehicle may leave depot d to drive to the start location of a trip, that is, the start time of the trip minus the driving time from the depot to the start location. Moreover, define H^d as the corresponding set of time points with $tp^{d1} < tp^{d2} < \cdots < tp^{dm}$. Furthermore, define st_i and et_i as respectively the start and ending time of trip i, $\text{trav}(r^d, i)$ and $\text{trav}(i, t^d)$ as the deadhead travel time from depot d to the start location of trip i and from the end location of trip i to depot d, respectively. For trip i, let parameter b_i^{dh} be equal to 1 if $st_i \le tp^{dh} < et_i$, and 0 otherwise. Similarly, let parameters

$$a_{ij}^{dh} = \begin{cases} 1, & \text{if } et_i \le tp^{dh} < st_j, \\ 0, & \text{otherwise,} \end{cases} \quad \text{for each arc } (i,j) \text{ with } i, j \in N^d,$$

$$a_{r_d j}^{dh} = \begin{cases} 1, & \text{if } st_j - \text{trav}(r_d, j) \le tp^{dh} < st_j, \\ 0, & \text{otherwise,} \end{cases} \quad \text{for each arc } (r_d, j) \text{ with } j \in N^d, \text{ and}$$

$$a_{i t_d}^{dh} = \begin{cases} 1, & \text{if } et_i \le tp^{dh} < et_i + \text{trav}(i, t_d), \\ 0, & \text{otherwise,} \end{cases} \quad \text{for each arc } (i, t_d) \text{ with } i \in N^d.$$

Using decision variables y_{ij}^d, $(i,j) \in A^{d*}$, with $y_{ij}^d = 1$ if a vehicle from depot d is assigned to arc (i,j), $y_{ij}^d = 0$ otherwise, variables x_i^d, $i \in N^d$, with $x_i^d = 1$ if trip i is assigned to a vehicle from depot d, $x_i^d = 0$ otherwise, and variables B^d to denote the number of vehicles used from depot $d \in D$, the S-MDVSP can be formulated as follows:

$$\text{Min} \sum_{d \in D} \left(cB^d + \sum_{(i,j) \in A^{d*}} c_{ij}^d y_{ij}^d \right), \tag{1}$$

$$\text{subject to} \sum_{\{i:(i,j) \in A^{d*}\}} y_{ij}^d - x_j^d = 0 \quad \forall d \in D,\ \forall j \in N^d, \tag{2}$$

$$\sum_{\{j:(i,j) \in A^{d*}\}} y_{ij}^d - x_i^d = 0 \quad \forall d \in D,\ \forall i \in N^d, \tag{3}$$

$$\sum_{d \in D_i} x_i^d = 1 \quad \forall i \in N, \tag{4}$$

$$\sum_{i \in N} b_i^{dh} x_i^d + \sum_{(i,j) \in A^{d*}} a_{ij}^{dh} y_{ij}^d \le B^d \quad \forall d \in D,\ \forall h \in H^d, \tag{5}$$

$$x_i^d \in \{0,1\} \quad \forall d \in D,\ \forall i \in N^d, \tag{6}$$

$$y_{ij}^d \in \{0,1\} \quad \forall d \in D,\ \forall (i,j) \in A^{d*}. \tag{7}$$

The objective function of this formulation is trivial. Recall, however, that we can ignore the long arcs because of the assumption that there are no costs involved when a vehicle is idle at the depot. Constraints (2) and (3) ensure that each trip is assigned to exactly one predecessor and one successor if this trip is assigned to depot d. Furthermore, constraints (4) ensure that every trip is assigned to exactly one depot. Constraints (5) state that the number of vehicles used from depot d should be at least the number of vehicles needed for trips and deadheads at any time point. It suffices to consider only time points in H^d because only at these time points can the number of vehicles in use increase; that is, a departure from the depot may occur. Moreover, if there are two consecutive time points in H^d between which no arrival at the depot can occur, then the number of vehicles at the latest time point is at least the number of vehicles at the earlier one. This means that the constraint for the earlier time point can be left out. Finally, the binary conditions on the x and y variables combined with constraints (5) guarantee that the variables B^d are positive integers.

3. Dynamic Vehicle Scheduling

In this section, we consider the following solution approach to the dynamic VSP: At certain moments, we construct a schedule for the next l time units, where we take into account those decisions already made earlier that cannot be changed anymore, and we also take into account in some way the future after the next l time units. In Figure 3, we show the same example as in Figure 2, where T is a point in time at which we construct a schedule for the period $[T, T+l)$. Note that we have drawn two nodes for every trip, namely, corresponding to the start and the end time of the

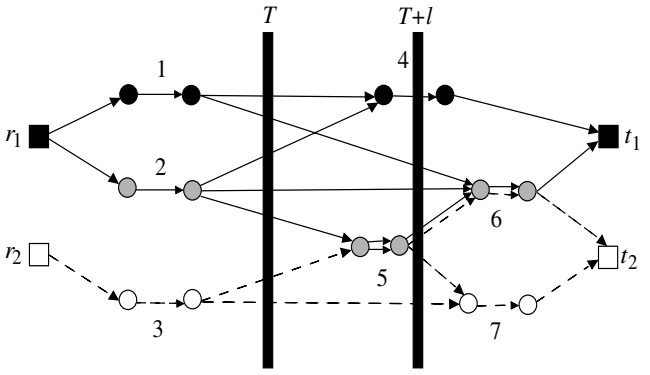

Figure 3 **Example—Network G^1 and G^2, Including Time Points T and $T+l$**

trip. We have already constructed the vehicle schedule up to time point T, so for that part of the schedule we only draw the arcs corresponding to the choices made. The decisions we have to make for the current period are which vehicle will do Trip 4 and which one, Trip 5. Of course, we can only choose vehicles from Depot 1 for Trip 4. When making these decisions, we already take into account, in some way, the network after $T+l$.

Our crucial assumption is that the travel times in the period $[T, T+l)$ are known with complete certainty. For the travel times after this period, we assume that we have information in the form of a number of possible scenarios, each with a certain probability of occurrence. These scenarios and the associated probabilities could be based on historical data, subjective expert opinions, or a combination of both. Note that one may choose to aggregate the scenarios into a single average scenario. Also note that in case no scenarios are available, one can still apply this approach, in which the travel times after the next l periods are simply made equal to the standard times that one also uses in the static problem.

In our current implementation, these scenarios are based on historical data, which means that every scenario corresponds to a day in the past. Therefore, the scenarios and the probabilities of these scenarios will not vary over the day. In principle, it is possible that the planners could modify these probabilities or the scenarios themselves during the day. For instance, on a rainy day, they can give higher probablities to scenarios corresponding to rainy days in the past.

After the $k \le l$ time units have passed, we repeat the above procedure, which means we construct a schedule for the period $[T+k, T+k+l)$. In our implementation, the lengths of the different periods are all equal except for the first one. However, for the approach itself the length of the periods can also vary over the day.

The optimization problem that we have to solve repeatedly is a stochastic programming problem if we explicitly consider multiple scenarios for the future travel times. If we consider a single scenario with average or standard travel times, we have to solve a sequence of static VSPs. In §3.1 and §3.2, we discuss the mathematical formulation and our solution method for the single-depot case. Finally, we discuss the multiple-depot case in §3.3.

3.1. Mathematical Formulation (Single Depot)

We will formulate the problem that we want to solve at time point T, where we schedule for the period $[T, T+l)$ and we use scenarios for the period after $T+l$. Let S denote the set of scenarios where possibly $|S| = 1$ (i.e., we also consider the case with a single scenario). We again denote by N the set of trips and we define for every scenario $s \in S$ a network $G^s = (N \cup r \cup t, A)$, where r and t are the source and the sink corresponding to the depot, respectively, and A is the set of arcs between two trips, from r to every trip and from every trip to t. Recall that st_i, $\text{trav}(r, i)$ and $\text{trav}(i, t)$ are defined as the start time of trip i, the deadhead travel time from the depot to the start location of trip i and from the end location of trip i to the depot, respectively. For the end time of trip i, we have to make a distinction between trips that end in the period $[T, T+l)$ and after $T+l$ for every scenario s. Define these end times as et_i^0 and et_i^s, respectively. Let $\text{trav}(i, j)$ be the deadhead travel time from the end location of trip i to the start location of trip j. Furthermore, denote by A^* the set of arcs without long arcs, by A_1 the subset of A^* that corresponds to the period $[T, T+l)$, which is the following set:

- (r, j), if $st_j - \text{trav}(r, j) \in [T, T+l)$,
- (i, j), if $st_j - \text{trav}(i, j) \in [T, T+l)$,
- (i, t), if $et_i^0 \in [T, T+l)$.

In the same way, we can define A_2 as the subset of arcs corresponding to the period after $T+l$. Furthermore, we define p_s as the probability of scenario s occurring and c'_{ij} and c^s_{ij} as the cost of arc (i, j) in period $[T, T+l)$ and the period after $T+l$ in scenario s. Here, the costs associated with an arc are a function of travel and idle time if the time between the trips i and j is nonnegative; otherwise it is a function of the delay or a sufficiently large number if delays are not allowed. Notice that by defining the cost in this way, we can use the same set of arcs for all scenarios. As before, we define c as the fixed vehicle cost.

As in §2, let tp^{sh} be the time points at which a vehicle may leave the depot to drive to the start location of a trip in scenario s, and define H^s as the corresponding set. Let

$$a^{sh}_{ij} = \begin{cases} 1, & \text{if } et^s_i \le tp^{sh} < st_j \\ -1, & \text{if } st_j \le tp^{sh} < et^s_i \\ 0, & \text{otherwise,} \end{cases} \quad \text{for each arc } (i, j) \text{ with } i, j \in N,$$

$$a^{sh}_{rj} = \begin{cases} 1, & \text{if } st_j - \text{trav}(r, j) \le tp^{sh} < st_j, \\ 0, & \text{otherwise,} \end{cases} \quad \text{for each arc } (r, j) \text{ with } j \in N, \quad \text{and}$$

$$a^{sh}_{it} = \begin{cases} 1, & \text{if } et^s_i \le tp^{sh} < et^s_i + \text{trav}(i, t), \\ 0, & \text{otherwise,} \end{cases} \quad \text{for each arc } (i, t) \text{ with } i \in N.$$

For arcs $(i, j) \in A_1$ similar definitions hold, but with et_i^0 instead of et_i^s. Furthermore, let b^{sh} be the number

of trips carried out at tp^{sh}; that is, we count all trips for which $st_i \le tp^{sh} < et_i^0$ for trips ending in $[T, T+l)$ and $st_i \le tp^{sh} < et_i^s$ otherwise. Note that if a trip starts late, the corresponding vehicle is counted twice in b^{sh}. The problem of double counting is solved by the definition of the a_{ij}^{sh} parameters because these are -1 in that case.

We use decision variables z_{ij} and y_{ij}^s, where $z_{ij} = 1$, if arc (i, j) is chosen in period $[T, T+l)$, $z_{ij} = 0$ otherwise and $y_{ij}^s = 1$, if arc (i, j) is chosen after $T+l$ in scenario s, $y_{ij}^s = 0$ otherwise. Furthermore, we also use B^s as a decision variable for the number of buses in scenario s. Then we get the following 0–1 program, where we minimize the expected vehicle and delay costs:

$$\text{Min } c\sum_{s\in S} p^s B^s + \sum_{(i,j)\in A_1} c'_{ij} z_{ij} + \sum_{s\in S} p^s \sum_{(i,j)\in A_2} c_{ij}^s y_{ij}^s, \quad (8)$$

subject to

$$\sum_{\{i:(i,j)\in A_1\}} z_{ij} + \sum_{\{i:(i,j)\in A_2\}} y_{ij}^s = 1 \qquad \forall s \in S, \forall j \in N, \quad (9)$$

$$\sum_{\{j:(i,j)\in A_1\}} z_{ij} + \sum_{\{j:(i,j)\in A_2\}} y_{ij}^s = 1 \qquad \forall s \in S, \forall i \in N, \quad (10)$$

$$b^{sh} + \sum_{(i,j)\in A_1} a_{ij}^{sh} z_{ij} + \sum_{(i,j)\in A_2} a_{ij}^{sh} y_{ij}^s \le B^s \qquad \forall s \in S, \forall h \in H^s, \quad (11)$$

$$z_{ij} \in \{0, 1\} \quad \forall (i, j) \in A_1, \quad (12)$$

$$y_{ij}^s \in \{0, 1\} \quad \forall s \in S, \forall (i, j) \in A_2. \quad (13)$$

Constraints (9) and (10) ensure that every trip has exactly one predecessor and one successor in every scenario. Furthermore, constraints (9) guarantee that if a trip i has a successor j and (i, j) is in set A_1, this holds for all scenarios. A similar remark holds for constraints (10) and the predecessor of a trip. Finally, constraints (11) and the fact that $c > 0$ guarantee that B^s is the number of vehicles in scenario s.

3.2. Solution Method (Single Depot)

For the single-depot VSP, a solution with possibly some trips starting late can be obtained by solving a sequence of problems (D-SDVSP-T) for different values of T to optimality. In Figure 4, we give a schematic overview of our solution method.

We use the CPLEX MIP solver to compute an optimal solution for problem D-SDVSP-T.

Step 0: Choose initial parameters T, k and l ($\geqslant k$).
Step 1: Solve problem D-SDVSP-T for period $[T, T+l)$
Update T: $T := T+k$.
Repeat Step 1 until the end of the day.

Figure 4 Solution Method for D-SDVSP

Step 1: Assign the trips to depots by solving S-MDVSP.
Step 2: For each depot, apply the dynamic approach of §3.2.

Figure 5 Solution Method for D-MDVSP

Notice that if we take a larger value for l, the assumptions are less realistic, because we assume that the travel times are known for the period $[T, T+l)$ at moment T. Furthermore, we found that, in our experiments, solving the LP-relaxation often resulted in an integer solution. Of course, if we have only one scenario, this problem is equivalent to the static SDVSP, which is known to be solvable in polynomial time.

3.3. Multiple-Depot Dynamic Vehicle Scheduling

For the dynamic approach to multiple-depot problems, we can formulate a stochastic programming problem that is a combination of S-MDVSP in §2 and D-SDVSP-T in §3.1. This formulation is actually given below. It turns out, however, that because of its size, it is hard to solve this formulation exactly in case of multiple scenarios. Therefore we use a so-called cluster-reschedule heuristic. In Figure 5, we give a schematic overview of the cluster-reschedule heuristic.

In the first step, we assign the trips to a certain depot, and in the second step, we solve the D-SDVSP for all depots. This means that a trip is assigned to a depot for the whole timetable period and that this cannot be changed anymore. For this reason, the heuristic is very attractive from a practical point of view because it is already known which subset of the trips vehicles from a certain depot may have to drive. In reality, where there are also drivers, this means that these drivers should only be educated about a subset of the trips.

Lower Bound. To evaluate the quality of the cluster-reschedule heuristic, we compute a lower bound using Lagrangian Relaxation. Therefore, we use the following formulation, which is a straightforward combination of S-MDVSP and D-SDVSP-T.

D-MDVSP-T:

$$\min \sum_{d\in D}\left(c\sum_{s\in S} p^s B^{ds} + \sum_{(i,j)\in A_1^d} c_{ij}^{d'} z_{ij}^d + \sum_{(i,j)\in A_2^d}\sum_{s\in S} p^s c_{ij}^{ds} y_{ij}^{ds}\right), \quad (14)$$

subject to

$$z_{ij}^d - y_{ij}^{ds} = 0 \qquad \forall d \in D,\ \forall s \in S,\ \forall (i, j) \in A_1^d, \quad (15)$$

$$\sum_{\{i:(i,j)\in A^{d*}\}} y_{ij}^{ds} - \sum_{\{k:(j,k)\in A^{d*}\}} y_{jk}^{ds} = 0 \qquad \forall d \in D,\ \forall s \in S,\ \forall j \in N, \quad (16)$$

$$\sum_{d\in D} \sum_{\{j:(i,j)\in A^{d*}\}} y_{ij}^{ds} = 1 \quad \forall s \in S,\ \forall i \in N, \quad (17)$$

$$\sum_{d\in D}\sum_{\{i:(i,j)\in A^{d*}\}} y_{ij}^{ds}=1 \quad \forall s\in S,\ \forall j\in N, \tag{18}$$

$$\sum_{i\in N} b_i^{hs}\sum_{\{j:(i,j)\in A^{d*}\}} y_{ij}^{ds}+\sum_{(i,j)\in A^{d*}} a_{ij}^{dhs}y_{ij}^{ds}\le B^{ds} \quad \forall d\in D,\ \forall s\in S,\ \forall h\in H^{ds}, \tag{19}$$

$$z_{ij}^{d}\in\{0,1\} \quad \forall d\in D,\ \forall (i,j)\in A_1^d, \tag{20}$$

$$y_{ij}^{ds}\in\{0,1\} \quad \forall d\in D,\ \forall s\in S,\ \forall (i,j)\in A^{d*}. \tag{21}$$

To compute a lower bound, we associate Lagrangian multipliers λ_{ij}^{ds} and μ_j^s with constraints (15) and (16), respectively. Furthermore, we define the corresponding vectors λ and μ. The Lagrangian subproblem becomes

$$\Phi(\lambda,\mu)=\Phi_y(\lambda,\mu)+\Phi_z(\lambda)$$

with

$$\Phi_y(\lambda,\mu)=\min \sum_{d\in D}\left(c\sum_{s\in S}p^sB^{ds}+\sum_{(i,j)\in A^{d*}}\sum_{s\in S}\bar{c}_{ij}^{ds}y_{ij}^{ds}\right),$$

subject to (17)–(19), (21),

and

$$\Phi_z(\lambda)=\min \sum_{d\in D}\sum_{s\in S}\sum_{(i,j)\in A_1^d}(c_{ij}^d+\lambda_{ij}^{ds})z_{ij}^d,$$

subject to (20),

where

$$\bar{c}_{ij}^{ds}=\begin{cases} p^sc_{ij}^{ds}-\lambda_{ij}^{ds}+\mu_j^{ds}-\mu_i^{ds} & \text{if } (i,j)\in A_1^d, i\in N, j\in N,\\ p^sc_{ij}^{ds}-\lambda_{ij}^{ds}+\mu_j^{ds} & \text{if } (i,j)\in A_1^d, i=r_d, j\in N,\\ p^sc_{ij}^{ds}-\lambda_{ij}^{ds}-\mu_i^{ds} & \text{if } (i,j)\in A_1^d, i\in N, j=t_d,\\ p^sc_{ij}^{ds}+\mu_j^{ds}-\mu_i^{ds} & \text{if } (i,j)\in A_2^d, i\in N, j\in N,\\ p^sc_{ij}^{ds}+\mu_j^{ds} & \text{if } (i,j)\in A_2^d, i=r_d, j\in N,\\ p^sc_{ij}^{ds}-\mu_i^{ds} & \text{if } (i,j)\in A_2^d, i\in N, j=t_d. \end{cases}$$

Let $\bar{y}(\lambda,\mu)$ and $\bar{z}(\lambda)$ denote optimal solutions corresponding to $\Phi_y(\lambda,\mu)$ and $\Phi_z(\lambda)$, respectively, for given λ and μ. Then $\bar{y}(\lambda,\mu)$ is obtained by solving a S-SDVSP for every scenario, and $\bar{z}(\lambda)$ is obtained by pricing out each variable; that is, for each $d\in D$ and $(i,j)\in A_1^d$, $\bar{z}_{ij}^d=1$ if $c_{ij}^d+\lambda_{ij}^{ds}\le 0$ and $\bar{z}_{ij}^d=0$ otherwise.

We use subgradient optimization to solve the Lagrangian dual problem $\max_{\lambda,\mu}\Phi(\lambda,\mu)$ approximately. In the subgradient optimization, we use as upper bound the value of the solution produced by the cluster-reschedule heuristic. It is worthwhile to note that we found that the gap between the upper and lower bound is already small when we did not use the subgradient algorithm, but just chose the Lagrangian multipliers equal to 0, which is equivalent to deleting the relaxed constraints. For the static MDVSP, Löbel (1997) also found that by deleting constraints (16) a good lower bound could be obtained.

4. Computational Experience

We have evaluated our approach by using data from Connexxion, the largest bus company in the Netherlands. In §4.1, we discuss the data and the results of the static vehicle scheduling problem and discuss the results of our dynamic approach in §4.2. All tests reported in this section are executed on a Pentium III 450 MHz personal computer (128 MB RAM).

4.1. Data Description and Results S-VSP

The data set consists of 1,104 trips and 4 depots in the area between Rotterdam, Utrecht, and Dordrecht, three large cities in the Netherlands. On a typical workday, there are a lot of traffic jams in this area, especially during rush hours (in the morning toward Rotterdam and Utrecht and in the afternoon in the opposite direction). Of course, most trips are also during these hours, which can be seen in Figure 6.

In this figure one can also see that the minimum number of vehicles will be determined during one hour in the morning peak (from 7 AM until 8 AM). Furthermore, it is important to note that not all trips are allowed to be driven by a vehicle from every depot. In fact, almost half of the trips can only be assigned to one depot and only a small number can be assigned to all depots. On average, a trip can be assigned to 1.71 depots. Furthermore, we have historical data concerning the travel times for a period of 10 days (2 weeks from Monday to Friday). These are only the travel times for trips, not for deadheads. Therefore, we implicitly consider the travel times for

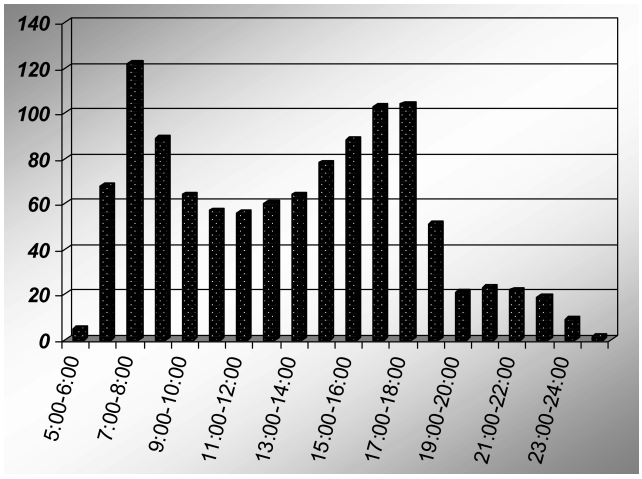

Figure 6 Distribution of Trips

Table 2 Optimal Solution of the S-VSP

Vehicle costs	1,102,538
Number of vehicles	109 (9-8-37-55)
CPU (sec.)	64

deadheads as fixed. Notice, however, that a similar approach can also be used if this is not the case. Furthermore, we assume that the actual travel time can never be less than the travel time in the timetable. This means that delays are always nonnegative and a bus is never too early at the end location. In practice, this will never happen if a driver just waits at each stop until it is time to depart.

We assume that a bus will leave exactly at the start time of a trip if possible. Furthermore, we assume that a trip's delay is independent of the actual starting time of this trip. This is realistic, because the frequencies at the different lines are quite low (e.g., every half hour or hour), which means that the number of passengers does not increase significantly if the trip starts late. This is in contrast with urban transport, where the frequencies are typically much higher, for example, every 10 minutes. Then, if a trip starts more than five minutes late, it gets an additional delay that depends on the actual starting time because there are more passengers at the different bus stops taking this trip who would have taken the next one if this trip had no delay.

We use 10,000 as fixed costs per vehicle and variable vehicle costs of 1 per minute that a vehicle is without passengers. If we solve this problem using the static VSP, we get the solution in Table 2.

The numbers between brackets show how the optimal number of vehicles is split over the depots. The computation time is about one minute when using the MIP solver of CPLEX, version 6.5, to solve model S-MDVSP in §2. Unfortunately, if we apply this optimal schedule on the 10 days for which we have historical data (realized travel times), a large number of trips will start late. In Table 3 we show, for these 10 days, the percentage of trips that start late and the cost of these delays, when we use as cost function $10x^2$, where x is the time in minutes that the trip starts late. We take a quadratic cost function, because a few small delays are preferred above one large delay. Furthermore, we scale it by a factor of 10 to get the delay costs in the same order of magnitude as the vehicle costs. This means that it is better to introduce one vehicle more if a delay of 32 minutes is prevented. As can be seen in the last column, on average 17.2% of the trips start late. Furthermore, one can see that there are many fewer delays on Fridays (Day 5 and 10), compared to the other days.

Table 4 Results of Introducing Fixed Buffer Times

Buffer time	No.	2 Min.	5 Min.	7 Min.	10 Min.
Vehicle costs	1,102,538	1,135,605	1,189,867	1,232,984	1,277,535
Number of vehicles	109	112	117	121	125
Trips late (%)	17.2	7.6	3.3	2.1	1.2
Delay costs	107,830	55,174	28,424	18,683	14,254

Of course, we can reduce the number of trips starting late and the delay costs by introducing fixed buffer times. The problem is still static, but we can take care of small delays. Table 4 shows the optimal solution, the average number of trips starting late and the average costs of these delays by using a fixed buffer time of 2, 5, 7, and 10 minutes after every trip, respectively. The detailed results for all different days are shown in Table 10 in the appendix.

It is obvious that the number of trips starting late and the delay costs can be reduced in this way; on the other hand, the number of vehicles and the vehicle costs increase enormously. However, even by introducing 5 or 10 minutes buffer time and thus using 8 respectively 16 vehicles more, there are still a significant number of trips starting late, and these are most of the time the large delays, which means that the delay costs are still quite high. In the next subsection, we show that our dynamic approach strongly outperforms the static approach.

4.2. Results D-VSP

In this subsection, we show the results of our approach to the dynamic vehicle scheduling problem and compare it with the results of the static case. We have used the following parameter settings.

- The first period starts when the first vehicle leaves the depot and ends at 7 AM. The length of the other periods are equal to l, where we vary the value of l (1, 5, 10, 15, 30, 60, and 120 minutes).
- Rescheduling only takes place at the start of a period ($k = l$), which means we follow a time-driven approach. Notice, however, that if $k = 1$, the approach coincides with an event-driven approach because the rescheduling takes place every time a delay occurs.

Table 3 Results when Using the Optimal Solution of the S-VSP

Day	1	2	3	4	5	6	7	8	9	10	Ave.
Trips late (%)	20.5	19.2	21.9	18.1	12.3	16.9	14.2	20.3	17.0	11.7	17.2
Costs	134,940	146,350	116,290	89,490	51,080	107,550	88,520	112,870	171,660	59,550	107,830

• With respect to the vehicle costs, the cost structure is the same as in the previous subsection.

• A cost of 10,000 is incurred for every trip starting late (independent of the size of the delay). For evaluation purposes, we still use the cost function defined in the previous subsection, but our first goal is to minimize the number of delays.

• We consider each of the 10 days for which we have historical data separately. In the case of using multiple scenarios (referred to as Case I), we took the realizations of the other 9 days as scenarios, where we gave one scenario (the same day but in the other week) a probability of 0.2 and the others a probability of 0.1. So if we optimize Day 1, we took as scenarios the realizations of Day 2 until Day 10, where Day 6 has a probability of 0.2 and the others 0.1.

• In the case of a single average scenario (referred to as Case II), we computed this average scenario by the weighted average of the nine scenarios as described above.

The average results over all days of the cluster-reschedule heuristic for D-VSP described in §3.3 are shown in Table 5. We show the results for the Cases I and II, as well as for the case where we do not use the historical data (Column 0) for different settings of l. In this table #V means the number of vehicles used, %L the percentage of trips starting late, and DC the cost of these delays.

The case where we do not use historical data and k and l equal 1 can be seen as an online approach where we know at the start of the day the travel times until 7 AM and modify the schedules every time a delay occurs. One can see that there are still a lot of delays in the case where we do not consider historical data. Therefore, we will focus in the remaining of the paper on Cases I and II. In Table 11 in the appendix, we give a more detailed overview of the results for these cases.

It is obvious that the quality of the results decreases if l decreases, because we assume that we know the realizations of the travel times l minutes in advance. In reality it is very difficult to predict these travel times very long before their actual realization. Furthermore, we can see that using more scenarios (Case I) leads to fewer delays and most often to fewer vehicles than Case II. So Case I clearly outperforms Case II. If we compare the results of Case I with the results in the previous subsection, one can see that for $l = 1$ the average number of vehicles used is 114.2, the average number of trips starting late is 2.1%, and the average delay costs are 8,960. These numbers are much better than the average results of the static case with a fixed buffer time of five minutes (see Table 4), which means that we clearly outperform the traditional static solution. Only by introducing a buffer time of 10 minutes do we get on average fewer trips starting late (1.2% instead of 2.1%), but 125 vehicles are needed every day instead of 114.2 on average, which is a significant difference. Although we did not optimize on the size of the delays, the delay costs are clearly lower than in the static case.

Table 5 Average Results D-VSP

	0			I			II		
l	#V	%L	DC	#V	%L	DC	#V	%L	DC
120	111.7	2.22	11,615	113.2	0.53	1,672	113.8	0.85	8,630
60	110.9	3.84	19,567	114.2	0.76	1,827	114.9	1.39	5,981
30	110.4	5.76	29,628	113.8	1.03	3,714	115.3	2.17	10,971
15	109.7	9.22	51,551	113.5	1.73	9,224	115.7	3.60	25,365
10	109.6	9.58	49,126	113.8	1.83	6,835	115.9	3.87	22,577
5	109.3	11.12	61,424	114.0	1.97	7,788	116.1	4.40	23,939
1	109.2	12.38	65,785	114.2	2.14	8,960	115.9	4.99	25,716

Table 6 Computation Times for D-VSP (Depot 4)

	I			II		
l	CPU	Iter.	Max.	CPU	Iter.	Max.
120	117	10	47	8	10	1
60	164	19	39	13	19	1
30	270	37	37	22	37	1
15	496	71	37	41	71	1
10	702	104	39	60	104	1
5	1,687	204	55	122	204	1
1	6,088	861	55	535	857	1

In Table 6, we show the computation time in seconds (Column CPU), the number of iterations (iter.), and the maximum computation time of one iteration in seconds (max.) for the largest depot. The computation time for the other depots is much smaller than for Depot 4.

Of course, the computation time for Case I is much higher than for Case II, but it is important to note that the computation time per iteration and per depot is always less than l, which means that this approach makes sense in practice.

Lower Bound. In this subsection, we evaluate our cluster-reschedule method for solving the D-VSP by comparing its results to lower bounds per iteration and to the situation where we have perfect information.

Because we use a heuristic in every iteration, it is interesting to compare the results with a lower bound on the optimal solution in each iteration (see §3.3). Therefore, Table 7 gives the relative gap between the lower and the upper bound in the first iteration, where the optimization problem is the most difficult one since it is the largest. Because the gaps are almost the same for the different days, we only show the minimum, maximum, and average gap over all days. Because the gap is reasonably small in the first iteration, this gives an indication that the difference between an exact algorithm and our cluster-reschedule heuristic is small in all iterations.

Table 7 Gap in the First Period

	I (%)	II (%)
Minimum	3.42	5.11
Maximum	3.60	5.95
Average	3.51	5.70

Of course, in case of perfect information, which means that we know all realizations of the travel times in advance, we get a lower bound on our problem. In this case, we can ensure that no trips start late and we thus only have to minimize the total vehicle costs. This can be done by a heuristic or exact method. In the first case we first cluster the trips by solving S-MDVSP like in the cluster-reschedule heuristic and then schedule the trips to optimality. The results are shown in Table 8.

If we compare the solution of Case I in Table 5 with the solution above in the case of using the same heuristic, the difference is very small. For example, on average we get 2.1% of the trips starting late; we save 0.3 vehicles if we obtain the relevant information only 1 minute in advance compared to the situation where we have full information. Furthermore, we can see that the difference between the heuristic and the optimal solution with perfect information is 3.5% on average, which is similar to the results shown in Table 7. This means that the solution of the cluster-reschedule heuristic is close to the optimal solution in every iteration.

Sensitivity Analysis. In Table 5, we see that the costs are lower for larger values of l. This is, of course, because we assume that the travel times are known (can be estimated without any error) at time point T for the period $[T, T + l)$. Especially for large values of l, this assumption may be unrealistic. Therefore, we have performed a sensitivity analysis by considering small deviations of the actual travel times from the estimated ones. In this analysis, we use the same schedules as generated in the experiments discussed before. These are considered to be based on estimated travel times. We evaluate the schedules with respect to actual travel times, which may differ slightly from the estimate ones. Note that evalution means the calculation of the number of trips starting late and the delay costs with the simulated travel times. The number of vehicles does not change. Furthermore, we note that the sensitivity analysis is carried out to test whether our approach is robust. This is different than in, for example, Daduna et al. (1993), where sensitivity analysis is carried out on static VSPs to show that small changes in the start time of the trips can improve the optimal solution significantly.

Table 8 Results of the Heuristic and Optimal Solution in the Case of Perfect Information

Day	1	2	3	4	5	6	7	8	9	10	Ave.
Heuristic	117	113	114	116	113	115	115	115	114	113	114.5
Optimal	111	110	110	112	110	110	111	112	110	110	110.6

Table 9 Average Results with Extra Perturbation $\epsilon \sim N(0, \sigma^2)$

		I		II	
l	σ	%L	DC	%L	DC
120	2	13.0	12,626.4	13.8	21,724.2
120	1	6.8	3,876.5	7.3	11,435.1
120	0.5	3.3	2,219.9	3.7	9,371.1
120	0.25	0.9	1,745.9	1.3	8,742.6
60	2	12.5	12,864.2	13.9	18,757.2
60	1	6.7	4,106.7	7.6	9,151.6
60	0.5	3.3	2,400.7	4.1	7,310.6
60	0.25	1.1	1,883.5	1.8	6,766.9
30	2	12.7	12,229.9	13.8	22,252.2
30	1	6.7	5,374.2	7.7	13,267.0
30	0.5	3.4	4,161.8	4.5	11,629.6
30	0.25	1.4	3,794.7	2.5	11,071.7

We have simulated four times 100 runs of actual travel times, where the actual travel time is drawn from a normal distribution with a mean equal to the estimated travel time and variance σ^2. Furthermore, we still assume that the total delay is nonnegative. In Table 9, we show the average number of trips starting late (%L) and the average delay costs (DC) for Cases I and II for different values of l and σ. Note that we take the average results over all days.

It is obvious that the number of trips starting late and the delay costs increase, but they are still much lower than in the static case. Of course, it is reasonable that the estimates of the travel times become better if l decreases. Recall that, when $l = 1$, 2.1% of the trips were starting late with a cost of 8,960, while we have better results for $l = 30$ and small deviations from the estimated travel times. Furthermore, if we compare the results in Tables 5 and 9, we can see that small perturbations have only a small impact on the performance of our method.

5. Conclusion

The results reported in the previous section show that we can reduce the number of trips starting late and the delay costs by using only a few vehicles more if we use our dynamic method instead of the traditional static one. The case where we use multiple scenarios to describe the future travel times clearly outperforms the case with a single average scenario. However, our method uses the fact that the travel times are known a certain time before realization. It is obvious that this is only realistic if this time is small, but even then our method clearly outperforms the static one. Furthermore, the impact of small deviations from the estimated travel time on the performance of our method is quite small.

It is very important that the optimization problem in every iteration of our method is solved quickly. Therefore, we have not used an exact approach but a cluster-reschedule heuristic, where we first cluster the trips using the static VSP and then dynamically reschedule the trips per depot. We have shown that the gap between this cluster-reschedule heuristic and a lower bound on the overall problem is quite small. Finally, we have shown that also the gaps between the solutions generated by our method and the optimal solutions with perfect information are reasonable.

For future research, we want to integrate the dynamic vehicle scheduling with crew scheduling such that the whole process can be done dynamically, which is necessary if this approach is to be used in practice.

Acknowledgments
The authors are very grateful to Connexxion for providing the data and support for this research. Furthermore, they would also like to thank two anonymous referees for their useful comments, which have improved this paper.

Appendix. Detailed Results

Table 10 Detailed Results of Introducing Fixed Buffer Times

BT	Day	1	2	3	4	5	6	7	8	9	10	Ave.
0	%L	20.5	19.2	21.9	18.1	12.3	16.9	14.2	20.3	17.0	11.7	17.2
	DC	134,940	146,350	116,290	89,490	51,080	107,550	88,520	112,870	171,660	59,550	107,830
2	%L	8.9	9.6	8.3	8.8	5.0	7.9	6.2	7.8	8.3	5.2	7.6
	DC	69,950	85,420	47,710	56,220	25,540	138,680	34,520	27,280	55,730	14,690	55,174
5	%L	3.7	4.4	4.2	4.1	1.5	4.2	2.5	2.9	3.6	1.7	3.3
	DC	18,800	58,860	25,030	23,540	3,280	93,650	11,420	8,220	36,200	5,330	28,424
7	%L	2.3	3.3	2.0	2.7	0.6	2.7	1.6	2.1	2.4	1.2	2.1
	DC	7,340	78,180	14,950	17,750	2,170	14,790	7,870	6,400	32,300	5,260	18,683
10	%L	1.7	2.4	1.2	1.3	0.2	1.5	1.2	1.1	1.3	0.6	1.2
	DC	8,180	86,500	9,090	7,420	260	6,460	7,220	3,260	12,940	1,210	14,254

Table 11 Detailed Results D-VSP

	I			II			I			II		
I	#V	%L	DC	#V	%L	DC	#V	%L	DC	#V	%L	DC
			Day 1						Day 6			
120	116	0.3	660	116	1.0	1,120	113	1.0	260	114	0.5	690
60	118	0.8	2,760	117	1.5	3,380	113	1.0	260	115	1.0	5,200
30	117	1.7	12,790	116	2.5	8,560	113	0.9	740	115	1.6	5,480
15	116	2.3	16,070	116	3.3	10,610	113	1.5	10,190	116	3.0	16,990
10	116	3.0	13,770	116	4.2	10,020	113	1.7	11,280	116	3.4	19,440
5	116	3.5	17,590	116	5.1	15,440	114	1.2	4,190	116	3.9	15,720
1	117	3.5	17,870	116	5.6	17,660	114	1.4	7,280	116	4.6	18,550
			Day 2						Day 7			
120	112	0.6	1,200	113	1.6	67,890	112	0.5	610	113	0.5	550
60	113	0.4	1,720	114	1.2	28,390	114	0.5	520	114	1.5	2,580
30	112	1.0	2,980	115	1.6	29,120	114	0.6	3,900	115	2.1	13,670
15	112	1.4	26,930	115	3.4	94,920	113	1.2	5,060	115	3.4	12,950
10	113	0.9	6,790	115	3.0	39,330	114	0.9	3,770	116	3.3	11,700
5	114	1.2	8,770	116	3.6	46,830	113	1.4	5,350	115	4.0	11,590
1	114	1.9	8,420	116	4.1	36,750	114	1.0	5,730	114	4.3	18,740
			Day 3						Day 8			
120	112	0.9	8,250	112	1.3	8,320	114	0.5	140	114	0.8	630
60	114	1.4	1,890	116	1.9	2,710	114	0.8	2,610	115	2.2	4,970
30	113	0.8	1,120	116	2.3	3,310	114	1.1	3,310	116	3.3	6,960
15	113	1.5	6,440	116	4.4	14,090	113	3.2	6,070	116	6.0	20,220
10	112	2.4	7,700	116	5.5	20,970	114	3.1	6,780	116	6.3	27,890
5	114	2.5	14,740	116	5.9	23,950	114	3.4	7,550	116	7.5	30,950
1	113	2.6	19,780	116	6.3	26,140	114	3.5	7,960	116	8.2	27,460

Table 11 (cont'd.)

	I			II			I			II		
I	#V	%L	DC	#V	%L	DC	#V	%L	DC	#V	%L	DC
			Day 4						Day 9			
120	114	0.8	4,210	114	1.6	5,680	114	0.0	0	115	0.1	10
60	115	1.0	4,160	114	2.4	6,650	114	0.2	170	115	0.7	830
30	115	1.9	6,010	116	3.0	6,820	113	0.8	1,250	115	2.1	25,820
15	114	2.1	8,300	116	4.0	7,180	113	1.4	2,020	116	3.3	58,860
10	115	2.5	5,660	116	4.1	8,050	114	1.1	1,600	116	3.4	65,380
5	114	2.5	7,950	117	4.3	7,170	113	1.4	2,330	117	4.0	62,650
1	115	3.0	9,250	116	5.4	17,700	114	1.6	2,680	117	4.9	65,770
			Day 5						Day 10			
120	113	0.5	1,260	114	0.6	1,310	112	0.2	130	113	0.4	100
60	114	0.4	850	114	0.7	1,580	113	1.2	3,330	115	0.9	3,520
30	114	0.7	2,520	114	1.4	5,570	113	0.7	2,520	115	1.7	4,400
15	114	1.4	7,900	115	2.5	11,100	114	1.3	3,260	116	2.6	6,730
10	114	1.6	8,210	116	3.1	16,460	113	1.0	2,790	116	2.4	6,530
5	114	1.5	6,990	116	3.4	18,400	114	1.2	2,420	116	2.6	6,690
1	113	2.0	8,400	116	3.5	21,580	114	0.9	2,230	116	2.8	6,810

References

Bertossi, A. A., P. Carraresi, G. Gallo. 1987. On some matching problems arising in vehicle scheduling models. *Networks* **17** 271–281.

Daduna, J. R., M. Mojsilovic, P. Schütze. 1993. Practical experiences using an interactive optimization procedure for vehicle scheduling. D.-Z. Du, P. M. Pardalos, eds. *Network Optimization Problems: Algorithms, Applications and Complexity*. World Scientific, Singapore, 37–52.

Desrosiers, J., Y. Dumas, M. M. Solomon, F. Soumis. 1995. Time constrained routing and scheduling. M. O. Ball, T. L. Magnanti, C. L. Monma, G. L. Nemhauser, eds. *Handbooks in Operations Research and Management Science*, Vol. 8. *Network Routing*, North-Holland, Amsterdam, The Netherlands, 35–139.

Freling, R., J. M. Pinto Paixão, A. P. M. Wagelmans. 2001. Models and algorithms for single-depot vehicle scheduling. *Transportation Sci.* **35** 165–180.

Haase, K., G. Desaulniers, J. Desrosiers. 2001. Simultaneous vehicle and crew scheduling in urban mass transit systems. *Transportation Sci.* **35** 286–303.

Ichoua, S., M. Gendreau, J.-Y. Potvin. 2000. Diversion issues in real-time vehicle dispatching. *Transportation Sci.* **34** 426–438.

Lamatsch, A. 1992. An approach to vehicle scheduling with depot capacity constraints. M. Desrochers, J.-M. Rousseau, eds. *Computer-Aided Transit Scheduling*. Springer Verlag, Berlin, Germany, 181–195.

Laporte, G., F. V. Louveaux. 1998. Solving stochastic routing problems with the integer L-shaped method. T. G. Crainic, G. Laporte, eds. *Fleet Management and Logistics*. Kluwer, Boston, MA, 159–167.

Löbel, A. 1997. Optimal vehicle scheduling in public transit. Unpublished doctoral dissertation, Technische Universität Berlin, Berlin, Germany.

Madsen, O. B. G., H. F. Ravn, J. M. Rygaard. 1995. A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities and multiple objectives. *Ann. Oper. Res.* **60** 193–208.

Mesquita, M., J. Paixão. 1999. Exact algorithms for the multiple-depot vehicle scheduling problem based on multicommodity network flow type formulations. N. H. M. Wilson, ed. *Computer-Aided Transit Scheduling*. Springer Verlag, Berlin, Germany, 223–246.

Powell, W. B., P. Jaillet, A. Odoni. 1995. Stochastic and dynamic networks and routing. M. O. Ball, T. L. Magnanti, C. L. Monma, G. L. Nemhauser, eds. *Handbooks in Operations Research and Management Science*, Vol. 8. *Network Routing*, North-Holland, Amsterdam, The Netherlands, 141–295.

Powell, W. B., M. T. Towns, A. Marar. 2000. On the value of optimal myopic solutions for dynamic routing and scheduling problems in the presence of noncompliance. *Transportation Sci.* **34** 67–85.

Ribeiro, C. C., F. Soumis. 1994. A column generation approach to the multiple-depot vehicle scheduling problem. *Operations Res.* **42** 41–52.

Sleator, D. D., R. E. Tarjan. 1985. Amortized efficiency of list update and paging rules. *Com. ACM* **28** 202–208.

Yen, J. W., J. R. Birge. 2002. A stochastic programming approach to the airline crew scheduling problem. Technical report, Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL.