A DYNAMIC APPROACH TO VEHICLE SCHEDULING

DENNIS HUISMAN, RICHARD FRELING AND ALBERT WAGELMANS

ERIM REPORT SERIES *RESEARCH IN MANAGEMENT*	
ERIM Report Series reference number	ERS-2001-35-LIS
Publication	June 2001
Number of pages	19
Email address corresponding author	huisman@few.eur.nl
Address	Erasmus Research Institute of Management (ERIM) Rotterdam School of Management / Faculteit Bedrijfskunde Erasmus Universiteit Rotterdam P.O. Box 1738 3000 DR Rotterdam, The Netherlands Phone: +31 10 408 1182 Fax: +31 10 408 9640 Email: info@erim.eur.nl Internet: www.erim.eur.nl

Bibliographic data and classifications of all the ERIM reports are also available on the ERIM website:
www.erim.eur.nl

ERASMUS **R**ESEARCH **I**NSTITUTE OF **M**ANAGEMENT

REPORT SERIES
RESEARCH IN MANAGEMENT

BIBLIOGRAPHIC DATA AND CLASSIFICATIONS

Abstract: This paper presents a dynamic approach to the vehicle scheduling problem. 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. In our dynamic approach, we solve a sequence of optimization problems, where we take into account different scenarios for future travel times. 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 a fast solution by standard optimization software. We report on the results of a computational study with real life data, in which we compare different variants of our approach and perform a sensitivity analysis with respect to deviations of the actual travel times from the estimated ones.

Library of Congress Classification (LCC)	5001-6182	Business
	5201-5982	Business Science
	HD 69.T54	Scheduling
Journal of Economic Literature (JEL)	M	Business Administration and Business Economics
	M 11	Production Management
	R 4	Transportation Systems
	C 69	Mathematical methods and programming: other
European Business Schools Library Group (EBSLG)	85 A	Business General
	260 K	Logistics
	240 B	Information Systems Management
	260 Q	Mathematics
Gemeenschappelijke Onderwerpsontsluiting (GOO)		
Classification GOO	85.00	Bedrijfskunde, Organisatiekunde: algemeen
	85.34	Logistiek management
	85.20	Bestuurlijke informatie, informatieverzorging
	85.03	Methoden en technieken, operations research
Keywords GOO	Bedrijfskunde / Bedrijfseconomie	
	Bedrijfsprocessen, logistiek, management informatiesystemen	
	Scheduling, reistijden, optimalistie	
Free keywords	Vehicle scheduling, stochastic programming, dynamic scheduling, stochastic traveltimes, public transport	

A Dynamic Approach to Vehicle Scheduling

Dennis Huisman, Richard Freling and Albert P.M. Wagelmans
Econometric Institute, Erasmus University Rotterdam,
P.O. Box 1738, NL-3000 DR Rotterdam, The Netherlands
E-mail: huisman@few.eur.nl

Abstract

This paper presents a dynamic approach to the vehicle scheduling problem. 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. In our dynamic approach, we solve a sequence of optimization problems, where we take into account different scenarios for future travel times. 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 a fast solution by standard optimization software. We report on the results of a computational study with real life data, in which we compare different variants of our approach and perform a sensitivity analysis with respect to deviations of the actual travel times from the estimated ones.

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. In Figure 1 we show 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 results in trips that start late.

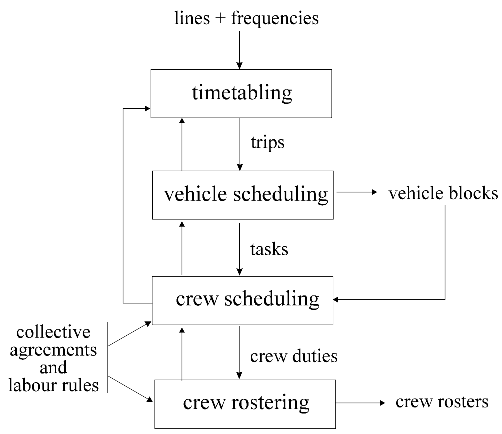

Figure 1: Traditional planning process

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 privatisation 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 only for a limited period. The contract specifies minimum service levels. In case these are not met, a fee is due and the contract may not be extended. 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 a dynamic approach for vehicle scheduling. The paper is organized as follows. Section 1 deals with the problem definition and discusses the potential benefit of using a dynamic approach for vehicle scheduling. In Section 2 we present a non-standard formulation of the static vehicle scheduling problem. This formulation will reappear in an extended form in Section 3, where we discuss our dynamic approach. We present results of computations with real life data from Connexxion in Section 4. In Section 5 we state our conclusions.

1 Vehicle Scheduling Problem

In this section, we discuss the vehicle scheduling problem and the potential benefit of a dynamic approach for solving this problem. Furthermore, we give a brief literature review about dynamic approaches.

1.1 Problem Definition

In the *(multiple-depot) vehicle scheduling problem* ((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;
- every depot has a maximum number of vehicles;
- some trips have to be assigned to vehicles from a certain set of depots;
- every vehicle has its own depot.

The vehicle costs consist of a fixed component for every vehicle and variable costs for idle and travel time. It is allowed that a vehicle returns to its own depot 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).

1.2 Potential Benefit of a Dynamic Approach

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 if 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 then 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 we should 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.*

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

Table 1: Data of the trips

The static optimal solution is that the trips 1 and 3 are assigned to vehicle 1 and vehicle 2 does trips 2 and 4.

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 5 minutes late if we use the schedule above. Even if we had added a buffer time of 5 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 6 minutes (or more), we would need three vehicles for doing these trips. Also note that *there is a solution for this example, where 2 vehicles are used and there are no trips starting late: vehicle 1 does the trips 1 and 4 and vehicle 2 trips 2 and 3.*

If the VSP is solved dynamically, which means that we reschedule a few times per day, we may be able to prevent the delays at the start of a trip in many occurrences.

1.3 Literature Review

To the best of our knowledge, there is no literature about dynamic approaches for the vehicle scheduling problem. Therefore, we only discuss here briefly some literature on solving other optimization problems using a dynamic approach. 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, they 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 crew scheduling and vehicle routing problems, would in reality lead to non-optimality. They argue that it is better to use algorithms that are more local in nature, e.g. greedy heuristics.

In the area of vehicle routing, there is a lot of literature about solving problems with random travel times. Most of the times stochastic programming is used to tackle the uncertainty (see e.g. Laporte and Louveaux (1998)). Recently, Yen and Birge (2000) have used stochastic programming to solve airline crew scheduling problems to get more robust schedules.

2 Static Vehicle Scheduling

In this section, we discuss the *static vehicle scheduling problem* (S-VSP) and we will present a non-standard 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. (1995) 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 for example Ribeiro and Soumis (1994)), but most of them are based on multicommodity flow formulations (see for example Löbel (1997) and Mesquita and Paixão (1999)). We will, however, give a formulation that is similar to the formulation proposed by Haase et al. (1999) in the context of integrated vehicle and crew scheduling.

Let D and $N = \{1, 2, ..., n\}$ 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 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 7 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.

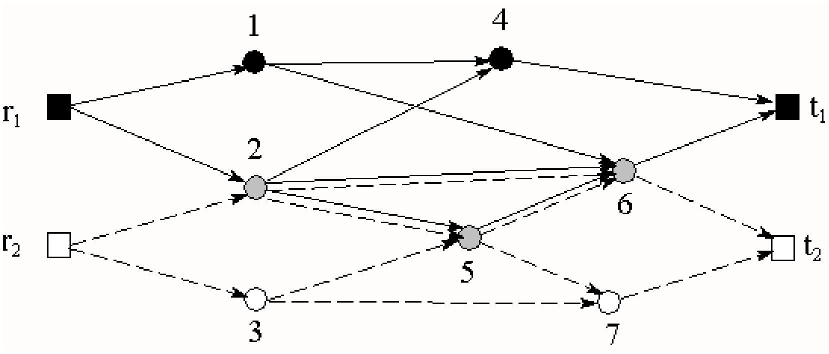

Figure 2: Example - Network G^1 and G^2

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. Furthermore, denote by k^d the number of vehicles that is available at depot d.

We could now give a typical multicommodity flow formulation of the (S-MDVSP). We will, however, present a different formulation that, in general, requires much less 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.

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

Define H^d as the set of time points at which a vehicle may leave depot d to drive to the start location of a trip, i.e., the start time of the trip minus the driving time from the depot to the start location. Furthermore, let $b_i^h = 1$ if trip i is carried out at time point $h \in H^d$, and $b_i^h = 0$ otherwise. Similarly, let $a_{ij}^{dh} = 1$ if time point $h \in H^d$ is between the start and ending time of the deadhead corresponding to arc $(i,j) \in A^{d*}$, and $a_{ij}^{dh} = 0$ otherwise.

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.

(S-MDVSP):

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

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

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

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

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

$$B^d \leq k^d \qquad \forall d \in D, \tag{6}$$

$$x_i^d \in \{0,1\} \qquad \forall d\in D, \forall i\in N^d, \tag{7}$$

$$y_{ij}^d \in \{0,1\} \qquad \forall d\in D, \forall (i,j)\in A^{d*}. \tag{8}$$

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) assure that each trip is assigned to exactly one predecessor and one successor if this trip is assigned to depot d. Furthermore, constraints (4) assure 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 $h \in H^d$. It suffices to consider only time points in H^d, since only at these time points the number of vehicles in use can increase, i.e., 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, constraints (6) assure that the number of vehicles used by a depot is not more than the number that is available.

3 Dynamic Vehicle Scheduling

In this section, we consider the following dynamic approach to the vehicle scheduling problem: at certain moments in time, we construct a schedule for the next l time units, where we take into account those decisions already made earlier that can not 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 where we construct a schedule for the period $[T, T+l)$. Note that we have drawn two nodes for every trip, namely one corresponding to the start and the end time of the 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 $T + l$.

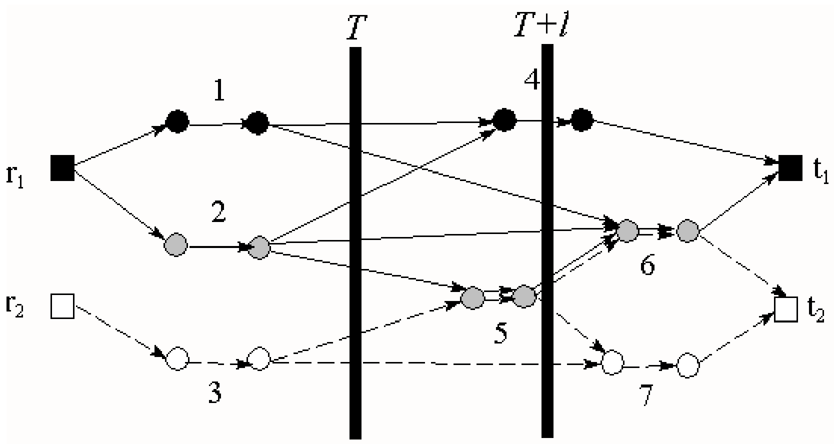

Figure 3: Example - Network G^1 and G^2 including the time points T and $T + l$

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, on 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 at all, one can still apply a dynamic approach in which the travel times after the next l periods are simply taken equal to the standard times that one also used in the static problem.

After the $k \leq l$ time units have passed, we repeat the above procedure.

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. In the case that we consider only a single scenario with average or standard travel times, we have to solve a sequence of static vehicle scheduling problems. In Subsections 3.1 and 3.2, we discuss the mathematical formulation and our solution method for the single-depot case, respectively. Finally, we discuss the multiple-depot case in Subsection 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 respectively the source and the sink corresponding to the depot and A is the set of arcs between two trips, from r to every trip and from every trip to t. 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 the start time of trip j minus the travel time from the depot to the start location of trip j is in the period $[T, T+l)$;
- (i, j), if the end time of trip j minus the travel time from the end location of trip i to the start location of trip j is in the period $[T, T+l)$;
- (i, t), if the end time of trip i is in the period $[T, T+l)$.

In the same way, we can define A_2 as the subset of arcs corresponding to the period after $T+l$. Also define p_s as the probability of scenario s occurring and define c'_{ij} and c^s_{ij} as the cost of arc (i, j) in respectively period $[T, T+l)$ and the period after $T+l$ in scenario s. Here, the cost of an arc is a function of travel and idle time if the time between the trips i and j is nonnegative and 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 and k as the number of vehicles that is available. Define H again as the set of time points at which a vehicle may leave the depot to drive to the start location of a trip. Let $a^{sh}_{ij} = 1$ if time point $h \in H$ is between the start and ending time of the deadhead corresponding to arc $(i, j) \in A^*$ in scenario s, and $a^{dh}_{ij} = 0$ otherwise. Notice that this is the same for all scenarios $s \in S$ in period $[T, T+l)$. Furthermore, let b^{sh} be the number of trips that is carried out at time point h in scenario s.

We use decision variables z_{ij} and y^s_{ij}, where $z_{ij} = 1$, if arc (i, j) is chosen in period $[T, T+l)$, $z_{ij} = 0$ otherwise and $y^s_{ij} = 1$, if arc (i, j) is chosen after $T+l$ in scenario s, $y^s_{ij} = 0$ otherwise. Furthermore, we also use B^s as 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.

(D-SDVSP-T):

$$
\begin{aligned}
\min\ c\sum_{s\in S} p^s B^s + \sum_{(i,j)\in A_1} c'_{ij} z_{ij} &\quad + \quad \sum_{s\in S} p^s \sum_{(i,j)\in A_2} c^s_{ij} y^s_{ij} && (9)\\
\text{subject to} \sum_{\{i:(i,j)\in A_1\}} z_{ij} + \sum_{\{i:(i,j)\in A_2\}} y^s_{ij} &\quad = \quad 1 \qquad \forall s \in S, \forall j \in N, && (10)\\
\sum_{\{j:(i,j)\in A_1\}} z_{ij} + \sum_{\{j:(i,j)\in A_2\}} y^s_{ij} &\quad = \quad 1 \qquad \forall s \in S, \forall i \in N, && (11)\\
b^{sh} + \sum_{(i,j)\in A_1} a^{sh}_{ij} z_{ij} + \sum_{(i,j)\in A_2} a^{sh}_{ij} y^s_{ij} &\quad \le \quad B^s \qquad \forall s \in S, \forall h \in H, && (12)\\
B^s &\quad \le \quad k \qquad \forall s \in S, && (13)\\
z_{ij} \in \{0,1\} & \qquad\qquad \forall (i,j) \in A_1, && (14)\\
y^s_{ij} \in \{0,1\} & \qquad\qquad \forall s \in S, \forall (i,j) \in A_2. && (15)
\end{aligned}
$$

Constraints (10) and (11) assure that every trip has exactly one predecessor and one successor in every scenario. Furthermore, constraint (10) guarantees 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 constraint (11) and the predecessor of a trip. Finally, constraint (12) and $c > 0$ guarantees that B^s is the number of vehicles in scenario s.

3.2 Solution Method (Single-Depot)

For the single depot vehicle scheduling problem, 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.

Step 0:	Choose initial parameters T, k and l $(\geq k)$.
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

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

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, if l is small we almost optimize in real time. 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 Section 2 and (D-SDVSP-T) in Subsection 3.1. This formulation is actually given in the next Subsection. 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.

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

Figure 5: Solution method for D-MDVSP

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, this heuristic is very attractive from a practical point of view.

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):

$$\begin{array}{rcll}
\min \sum_{d\in D}(c\sum_{s\in S} p^s B^{ds} + \sum_{(i,j)\in A_1^d} c_{ij}^{d'} z_{ij}^d & + & \sum_{(i,j)\in A_2^d}\sum_{s\in S} p^s c_{ij}^{ds} y_{ij}^{ds}) & (16)\\
\text{subject to} \qquad z_{ij}^d - y_{ij}^{ds} & = & 0 \quad \forall d\in D, \forall s\in S, \forall (i,j)\in A_1^d, & (17)\\
\sum_{\{i:(i,j)\in A^{d*}\}} y_{ij}^{ds} - \sum_{\{k:(j,k)\in A^{d*}\}} y_{jk}^{ds} & = & 0 \quad \forall d\in D, \forall s\in S, \forall j\in N, & (18)\\
\sum_{d\in D}\sum_{\{j:(i,j)\in A^{d*}\}} y_{ij}^{ds} & = & 1 \quad \forall s\in S, \forall i\in N, & (19)\\
\sum_{d\in D}\sum_{\{i:(i,j)\in A^{d*}\}} y_{ij}^{ds} & = & 1 \quad \forall s\in S, \forall j\in N, & (20)\\
\sum_{i\in N} b_i^{hs} \sum_{\{j:(i,j)\in A^{d*}\}} y_{ij}^{ds} + \sum_{(i,j)\in A^{d*}} a_{ij}^{dhs} y_{ij}^{ds} & \leq & B^{ds} \quad \forall d\in D, \forall s\in S, \forall h\in H^d, & (21)\\
B^{ds} & \leq & k^d \quad \forall d\in D, \forall s\in S, & (22)\\
z_{ij}^d & \in & \{0,1\} \quad \forall d\in D, \forall (i,j)\in A_1^d, & (23)\\
y_{ij}^{ds} & \in & \{0,1\} \quad \forall d\in D, \forall s\in S, \forall (i,j)\in A^{d*}. & (24)
\end{array}$$

To compute a lower bound, we relax the constraints (22) by deleting them and we associate Lagrangian multipliers λ_{ij}^{ds} and μ_j^s with constraints (17) and (18), 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

$$\begin{array}{rcl}
\Phi_y(\lambda,\mu) & = & \min \sum_{d\in D}(c\sum_{s\in S} p^s B^{ds} + \sum_{(i,j)\in A^{d*}}\sum_{s\in S} \overline{c}_{ij}^{ds} y_{ij}^{ds})\\
 & & \text{subject to (19)-(21), (24),}
\end{array}$$

and

$$\begin{array}{rcl}
\Phi_z(\lambda) & = & \min \sum_{d\in D}\sum_{s\in S}\sum_{(i,j)\in A_1^d} (c_{ij}^d + \lambda_{ij}^{ds}) z_{ij}^d\\
 & & \text{subject to (23),}
\end{array}$$

where

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

Let $\overline{y}(\lambda,\mu)$ and $\overline{z}(\lambda)$ denote optimal solutions corresponding to $\Phi_y(\lambda,\mu)$ and $\Phi_z(\lambda)$, respectively, for given λ and μ. Then $\overline{y}(\lambda,\mu)$ is obtained by solving a S-SDVSP for every scenario, and $\overline{z}(\lambda)$ is obtained by pricing out each variable, that is, for each $d \in D$ and $(i,j) \in A_1^d$, $\overline{z}_{ij}^d = 1$ if $c_{ij}^d + \lambda_{ij}^{ds} \leq 0$ and $\overline{z}_{ij}^d = 0$ 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. However, the gap between the upper and lower bound is already small if we do not use the subgradient algorithm, but just choose the Lagrangian multipliers equal to 0, which is equivalent to deleting the relaxed constraints.

4 Computational Experience

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

4.1 Data Description and Results S-VSP

In this subsection, we discuss some important properties of the data set. The set consists of 1104 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 towards 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 very small number can be assigned to all depots. On average, a trip can be assigned to 1.71 depots. We do not consider a maximum number of vehicles per depot, but we assume that there are always enough vehicles at the 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 and not for deadheads. Therefore, we implicitly consider the travel times for deadheads as fixed. Notice, however, that a similar approach can also be used if this is not

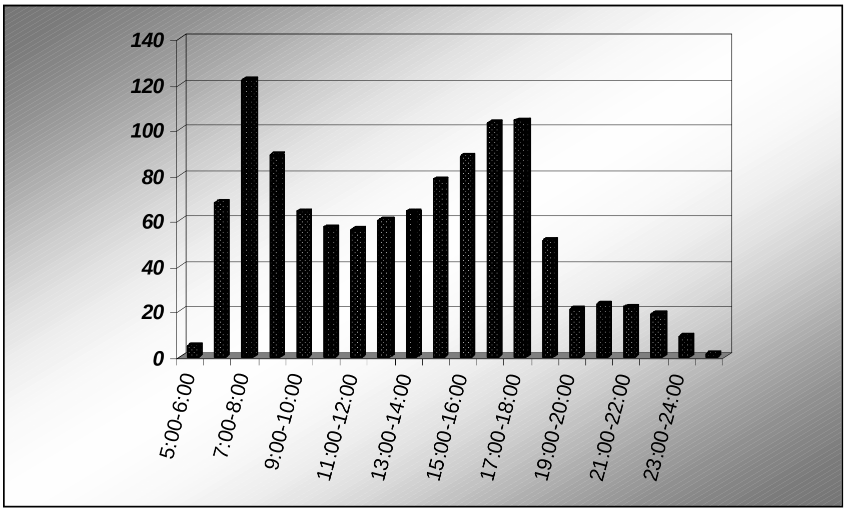

Figure 6: Distribution of the trips over the day

the case. Furthermore, we assume that the actual travel time of a trip 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 this is possible. Furthermore, we assume that the delay of a trip 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 in contrast with urban transport, where the frequencies are typically much higher, e.g. every 10 minutes. Then, if a trip starts more than 5 minutes late, it gets an additional delay that depends on the actual starting time.

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

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

Table 2: Optimal solution of the S-VSP

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 Section 2. Unfortunately, if we use this optimal schedule in practice, i.e. if we apply it on the days for which we have historical data available, a large number of trips will start late. In Table 3, we show, for the 10 realizations, 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 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 is starting late. Furthermore, one can see that there are much less delays on Friday's (day 5 and 10) compared to the other days.

day	1	2	3	4	5	6	7	8	9	10	Av.
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

Table 3: Realizations by using the optimal solution of the S-VSP

Of course, we can reduce these numbers 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 respectively 2 and 5 minutes after every trip.

buffer time	no	2 min.	5 min.
vehicle costs	1,102,538	1,135,605	1,189,867
number of vehicles	109	112	117
trips late (%)	17.2	7.6	3.3
delay costs	107,830	55,174	28,424

Table 4: Results of introducing fixed buffer times

It is obvious that the number of trips starting late and the delay costs can be reduced in this way and that the number of vehicles and the vehicle costs increase. But even by introducing 5 minutes buffer time and thus using 8 vehicles more, there is 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 these results.

4.2 Results D-VSP

In this subsection, we show the results of our dynamic approach 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 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 ($k = l$).

- 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 I), we took the realizations of the other 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 II), we computed this average scenario by the weighted average of the 9 scenarios as described above.

The average results over all days of the cluster-reschedule heuristic for D-VSP described in Subsection 3.3 are shown in Figure 7. We show the results for the cases I and II, as well for the case where we do not use the historical data (column 0) for different settings of l. In this figure #V means the number of vehicles used, %L the percentage of trips starting late and DC the cost of these delays.

	Average								
	0			I			II		
l	#V	%L	DC	#V	%L	DC	#V	%L	DC
120	112.4	2.03	10639	113.3	0.49	1064	113.9	0.81	8022
60	112	3.74	18558	114.5	0.72	1743	115.2	1.36	6780
30	111.7	5.69	29372	114.1	1.08	4222	115.6	2.19	11458
15	111.5	9.20	51067	114	1.72	9528	115.8	3.60	25780
10	111.4	9.47	47578	114.4	1.75	6239	116.3	3.67	22358
5	111.2	11.04	64726	114.6	1.99	8214	116.6	4.44	25450
1	111.5	12.23	65431	115.1	2.10	8723	116.6	5.02	27416

Figure 7: Average results D-VSP

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 the cases I and II. In Figure 8, 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. Furthermore, we can see that using more scenarios (case I) leads to less delays and most often to less 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 the maximum number of vehicles used is 117 and the maximum number of trips starting late is 3.5% (both for day 1 and l is 1). These numbers are almost equal to the average results of the static case with a fixed buffer time of 5 minutes (see Table 4), which means that we clearly outperform the traditional

	Day 1						Day 6					
	I			II			I			II		
l	#V	%L	DC	#V	%L	DC	#V	%L	DC	#V	%L	DC
120	116	0.3	660	116	1.0	1120	113	1.0	260	114	0.5	690
60	118	0.6	2420	117	1.4	4250	113	1.0	260	115	1.0	5200
30	117	1.6	12540	116	2.4	9220	113	0.9	740	115	1.6	5480
15	117	2.0	15500	116	3.2	11480	113	1.5	10190	116	3.0	16990
10	117	2.4	13010	116	4.2	10020	113	1.7	9110	116	3.4	19440
5	117	3.4	17650	116	4.9	15340	114	1.2	4190	117	3.9	15720
1	117	3.5	18150	116	5.5	17560	116	1.4	7280	117	4.6	18550

	Day 2						Day 7					
	I			II			I			II		
l	#V	%L	DC	#V	%L	DC	#V	%L	DC	#V	%L	DC
120	112	0.6	1200	113	1.6	67890	112	0.5	610	113	0.5	550
60	113	0.4	1720	114	1.2	28390	114	0.5	520	115	1.5	2610
30	113	1.0	2260	115	1.6	29120	114	0.8	5590	116	2.1	13670
15	113	1.4	23480	115	3.4	94920	114	1.2	6660	116	3.4	12950
10	113	0.9	3540	115	3.0	39330	114	1.0	4660	117	1.6	11700
5	114	1.5	6460	117	4.3	63990	114	1.4	5750	116	4.0	11600
1	115	2.0	5060	117	4.7	53900	115	1.3	6630	116	4.5	18320

	Day 3						Day 8					
	I			II			I			II		
l	#V	%L	DC	#V	%L	DC	#V	%L	DC	#V	%L	DC
120	112	0.8	2860	112	1.2	2930	114	0.4	100	114	0.8	630
60	114	1.3	1640	115	1.8	9650	115	0.8	2610	116	2.2	5610
30	113	0.8	1120	116	2.3	3310	115	1.1	3310	117	3.3	7600
15	113	1.5	9020	116	4.3	13840	114	3.0	5670	116	6.0	20430
10	113	2.2	7770	116	5.3	20470	115	3.1	6780	116	6.2	26450
5	113	2.5	21440	116	5.7	23700	115	3.5	7590	116	7.7	29530
1	113	2.5	20670	116	6.0	23660	116	3.3	7880	117	8.3	26390

	Day 4						Day 9					
	I			II			I			II		
l	#V	%L	DC	#V	%L	DC	#V	%L	DC	#V	%L	DC
120	115	0.6	3560	115	1.4	4990	114	0.0	0	115	0.1	10
60	117	0.9	3910	116	2.3	6160	114	0.2	170	115	0.7	830
30	116	2.1	8610	117	3.2	9420	113	1.0	2220	115	2.3	26790
15	115	2.3	10900	116	4.1	9780	113	1.6	2480	116	3.5	59830
10	116	2.5	5660	118	4.1	8050	115	1.1	1600	117	3.4	65380
5	116	2.3	5500	118	4.2	7130	114	1.4	2570	118	4.0	62650
1	116	2.8	9070	117	5.3	17660	115	1.6	2680	118	4.9	69730

	Day 5						Day 10					
	I			II			I			II		
l	#V	%L	DC	#V	%L	DC	#V	%L	DC	#V	%L	DC
120	113	0.5	1260	114	0.6	1310	112	0.2	130	113	0.4	100
60	114	0.4	850	114	0.7	1580	113	1.2	3330	115	0.9	3520
30	114	0.7	2980	114	1.4	5570	113	0.8	2850	115	1.7	4400
15	114	1.4	8120	115	2.5	11100	114	1.3	3260	116	2.5	6480
10	114	1.4	7440	116	3.1	16460	114	1.0	2820	116	2.3	6280
5	114	1.5	8570	116	3.4	18400	115	1.2	2420	116	2.4	6440
1	113	1.6	7580	116	3.5	21580	115	0.9	2230	116	2.8	6810

Figure 8: Results D-VSP

static solution. Although, we did not optimize on the size of the delays, also the delay costs are clearly lower than in the static case.

In Figure 9, 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.

	I			II		
	Depot 4			Depot 4		
l	CPU	iter.	max	CPU	iter.	max
120	128	10	56	7	10	1
60	168	19	46	11	19	1
30	278	37	45	21	37	1
15	934	71	75	77	71	2
10	760	104	56	56	104	1
5	1271	204	46	108	204	1
1	5967	861	56	508	857	1

Figure 9: Computation times for D-VSP

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 its results with a lower bound on the optimal solution in each iteration (see Subsection 3.3). Therefore, we give in Figure 10 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. Since the gap is reasonable 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.

	I	II
	gap	gap
minimum	3.42%	5.11%
maximum	3.60%	5.95%
average	3.51%	5.70%

Figure 10: Upper and lower bound in the first period.

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 are starting late and thus we 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 using solving (S-MDVSP) like in the cluster-reschedule heuristic and then schedule the trips optimally. The results are shown in Figure 11.

Day	1	2	3	4	5	6	7	8	9	10	av.
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

Figure 11: Results of the heuristic and the optimal solution in the case of perfect information

If we compare the solution of case I in Figures 7 and 8 with the solution above in the case of using the same heuristic as very small. For example, on average we get only 1.1 vehicles more while 2.1% of the trips is starting late if we get 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 optimal solution with perfect information is on average 3.5 %, which is similar as in Figure 10. This means that the solution of the cluster-reschedule heuristic is close to the optimal solution in every iteration.

Sensitivity analysis

In Figure 7, 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. We have simulated four times 100 runs of actual travel times, where the actual travel time is drawn from a normal distribution with mean equal to the estimated travel time and variance σ^2. Furthermore, we still assume that the total delay is nonnegative. In Figure 12, we show the average number of trips starting late (%L) and the average delay costs (DC) for case 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 less than in the static case. Of course, it is reasonable that the estimates of the travel times become better if l decreases. Recall that, for l equal to 1, 3.5% of the trips were starting late with a cost of 18,150, while we have better results for l equal to 30 and small deviations from the estimated travel times. Furthermore, if we compare the results of the Figures 7 and 12, one 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 at the price of 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 only a single

		I		II	
l	σ	%L	DC	%L	DC
120	2	13.0	11814.9	13.7	20844.1
120	1	6.8	3194.5	7.3	10737.1
120	0.5	3.2	1568.4	3.7	8723.3
120	0.25	0.9	1124.6	1.2	8126.8
60	2	12.5	12759.0	13.9	18799.7
60	1	6.6	4034.3	7.6	9249.1
60	0.5	3.3	2329.6	4.1	7407.0
60	0.25	1.1	1813.0	1.8	6862.3
30	2	12.4	12817.8	13.8	22715.0
30	1	6.7	5916.6	7.7	13758.8
30	0.5	3.4	4684.8	4.5	12119.6
30	0.25	1.4	4289.4	2.5	11558.3

Figure 12: Average results with extra perturbation $\epsilon \sim N(0, \sigma^2)$

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 quite fast. 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 we 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 used in practice.

Acknowledgments The authors are very grateful to Connexxion for providing the data and their support of this research.

References

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

[2] R. Freling, J.M. Pinto Paixão, and A.P.M. Wagelmans. Models and algorithms for vehicle scheduling. Technical Report 9562/A, Econometric Institute, Erasmus University Rotterdam, Rotterdam, 1995. To appear in Transportation Science.

[3] K. Haase, G. Desaulniers, and J. Desrosiers. Simultaneous vehicle and crew scheduling in urban mass transit systems. Technical Report GERAD G-98-58, École des Hautes Études Commerciales, Montréal, 1999.

[4] G. Laporte and F.V. Louveaux. Solving stochastic routing problems with the integer l-shaped method. In T.G. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, pages 159–167. Kluwer, 1998.

[5] A. Löbel. *Optimal Vehicle Scheduling in Public Transit.* PhD thesis, Technische Universität Berlin, 1997.

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

[7] W.B. Powell, P. Jaillet, and A. Odoni. Stochastic and dynamic networks and routing. In C. L. Monma M. O. Ball, T. L. Magnanti and G. L. Nemhauser, editors, *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*, pages 141–295. North-Holland, 1995.

[8] W.B. Powell, M.T. Towns, and A. Marar. On the value of optimal myopic solutions for dynamic routing and scheduling problems in the presence of noncompliance. *Transportation Science*, 34(1):67–85, 2000.

[9] C.C. Ribeiro and F. Soumis. A column generation approach to the multiple-depot vehicle scheduling problem. *Operations Research*, 42:41–52, 1994.

[10] J.W. Yen and J.R. Birge. A stochastic programming approach to the airline crew scheduling problem. World Wide Web, http://www.mcs.vuw.ac.nz/opre/orsnz2000/papers/JoyceYen.pdf, 2000.

Publications in the Report Series Research* in Management

ERIM Research Program: "Business Processes, Logistics and Information Systems"

2001

Bankruptcy Prediction with Rough Sets
Jan C. Bioch & Viara Popova
ERS-2001-11-LIS

Neural Networks for Target Selection in Direct Marketing
Rob Potharst, Uzay Kaymak & Wim Pijls
ERS-2001-14-LIS

An Inventory Model with Dependent Product Demands and Returns
Gudrun P. Kiesmüller & Erwin van der Laan
ERS-2001-16-LIS

Weighted Constraints in Fuzzy Optimization
U. Kaymak & J.M. Sousa
ERS-2001-19-LIS

Minimum Vehicle Fleet Size at a Container Terminal
Iris F.A. Vis, René de Koster & Martin W.P. Savelsbergh
ERS-2001-24-LIS

The algorithmic complexity of modular decompostion
Jan C. Bioch
ERS-2001-30-LIS

A Dynamic Approach to Vehicle Scheduling
Dennis Huisman, Richard Freling & Albert Wagelmans
ERS-2001- 35-LIS

Effective Algorithms for Integrated Scheduling of Handling Equipment at Automated Container Terminals
Patrick J.M. Meersmans & Albert Wagelmans
ERS-2001-36-LIS

Rostering at a Dutch Security Firm
Richard Freling, Nanda Piersma, Albert P.M. Wagelmans & Arjen van de Wetering
ERS-2001-37-LIS

2000

A Greedy Heuristic for a Three-Level Multi-Period Single-Sourcing Problem
H. Edwin Romeijn & Dolores Romero Morales
ERS-2000-04-LIS

---

* A complete overview of the ERIM Report Series Research in Management:
http://www.ers.erim.eur.nl

ERIM Research Programs:
LIS Business Processes, Logistics and Information Systems
ORG Organizing for Performance
MKT Marketing
F&A Finance and Accounting
STR Strategy and Entrepreneurship

Integer Constraints for Train Series Connections
Rob A. Zuidwijk & Leo G. Kroon
ERS-2000-05-LIS

Competitive Exception Learning Using Fuzzy Frequency Distribution
W-M. van den Bergh & J. van den Berg
ERS-2000-06-LIS

Models and Algorithms for Integration of Vehicle and Crew Scheduling
Richard Freling, Dennis Huisman & Albert P.M. Wagelmans
ERS-2000-14-LIS

Managing Knowledge in a Distributed Decision Making Context: The Way Forward for Decision Support Systems
Sajda Qureshi & Vlatka Hlupic
ERS-2000-16-LIS

Adaptiveness in Virtual Teams: Organisational Challenges and Research Direction
Sajda Qureshi & Doug Vogel
ERS-2000-20-LIS

Assessment of Sustainable Development: a Novel Approach using Fuzzy Set Theory
A.M.G. Cornelissen, J. van den Berg, W.J. Koops, M. Grossman & H.M.J. Udo
ERS-2000-23-LIS

Applying an Integrated Approach to Vehicle and Crew Scheduling in Practice
Richard Freling, Dennis Huisman & Albert P.M. Wagelmans
ERS-2000-31-LIS

An NPV and AC analysis of a stochastic inventory system with joint manufacturing and remanufacturing
Erwin van der Laan
ERS-2000-38-LIS

Generalizing Refinement Operators to Learn Prenex Conjunctive Normal Forms
Shan-Hwei Nienhuys-Cheng, Wim Van Laer, Jan Ramon & Luc De Raedt
ERS-2000-39-LIS

Classification and Target Group Selection bases upon Frequent Patterns
Wim Pijls & Rob Potharst
ERS-2000-40-LIS

Average Costs versus Net Present Value: a Comparison for Multi-Source Inventory Models
Erwin van der Laan & Ruud Teunter
ERS-2000-47-LIS

Fuzzy Modeling of Client Preference in Data-Rich Marketing Environments
Magne Setnes & Uzay Kaymak
ERS-2000-49-LIS

Extended Fuzzy Clustering Algorithms
Uzay Kaymak & Magne Setnes
ERS-2000-51-LIS

Mining frequent itemsets in memory-resident databases
Wim Pijls & Jan C. Bioch
ERS-2000-53-LIS

Crew Scheduling for Netherlands Railways. "Destination: Curstomer"
Leo Kroon & Matteo Fischetti
ERS-2000-56-LIS