

A Solution Approach for Dynamic Vehicle and Crew Scheduling

Dennis Huisman* and Albert P.M. Wagelmans

Erasmus Center for Optimization in Public Transport (ECOPT) &

Econometric Institute, Erasmus University Rotterdam,

P.O. Box 1738, NL-3000 DR Rotterdam, the Netherlands

E-mail: huisman@few.eur.nl - wagelmans@few.eur.nl

Econometric Institute Report EI2004-02

Abstract

In this paper, we discuss the dynamic vehicle and crew scheduling problem and we propose a solution approach consisting of solving a sequence of optimization problems. Furthermore, we explain why it is useful to consider such a dynamic approach and compare it with a static one. Moreover, we perform a sensitivity analysis on our main assumption that the travel times of the trips are known exactly a certain amount of time before actual operation.

We provide extensive computational results on some real-world data instances of a large public transport company in the Netherlands. Due to the complexity of the vehicle and crew scheduling problem, we solve only small and medium-sized instances with such a dynamic approach. We show that the results are good in the case of a single depot. However, in the multiple-depot case, the dynamic approach does not perform so well. We investigate why this is the case and conclude that the fact that the instance has to be split in several smaller ones, has a negative effect on the performance.

Keywords: transportation, large-scale optimization, dynamic planning, vehicle and crew scheduling

1 Introduction

Due to privatization and the growing competition in the public transport market, it has become much more important for public transport companies to provide

*Corresponding Author

an adequate service level to their customers. For instance, in the Netherlands, public transport companies (will) sign a contract with the government to provide transport in a certain area that is only valid for a limited period. The contract specifies minimum service levels. In case these are not met, a penalty 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 Huisman et al. (2001) we have introduced a new approach to vehicle scheduling. Vehicle scheduling is the process of assigning trips to vehicles. These trips result from the timetable and are given. In the paper, we looked at a dynamic method instead of the traditional static one. That is, a vehicle schedule is not constructed for a whole period, but is generated online. For example, it can be generated every hour for the next one. The results of that approach were very promising. The number of trips starting late was reduced at the price of using only a few vehicles more. Furthermore, we considered two different cases, one with a single scenario representing the average travel time and one with multiple scenarios for future travel times. The second case clearly outperformed the first one. However, our method used the fact that the travel times are known exactly a certain time before realization. It is obvious that this is only realistic if this time is small, but even then our method clearly outperformed the static one. Furthermore, the impact of small deviations from the estimated travel times on the performance of our method was quite small. In such an online approach, it is very important that the optimization problem in every iteration is solved quite fast. Therefore, we did not use an exact approach, but a cluster-reschedule heuristic, where we first cluster the trips and assign them to different depots using the static vehicle scheduling problem, and then dynamically reschedule the trips per depot. We showed that the gap between this cluster-reschedule heuristic and a lower bound on the overall problem is less than four percent, which is reasonably small.

In this paper, we will integrate the dynamic vehicle scheduling problem with crew scheduling, i.e. assigning trips to crews, such that the whole process can be done dynamically, which is necessary if such a dynamic approach is used in practice. Furthermore, we compare this with the approach of static vehicle and crew scheduling with buffer times, which is the approach currently used in practice. The static as well as the integrated problem can be solved either sequentially, i.e. first vehicle scheduling and then crew scheduling, or in an integrated way. Notice that solving the integrated vehicle and crew scheduling problem asks much

more computational power than the vehicle scheduling problem. Therefore, for the integrated problem, dynamic approaches can only be used if the underlying problems are solved heuristically, which is likely to have a negative effect on the quality. However, such a negative effect may be compensated by the positive effect of using a dynamic approach.

Of course, delays are not only an important issue in bus transport. In the airline world, a related problem, disruption management, is one of the major issues nowadays (see Horner (2002) about the success of Operations Research after September 11). In the disruption management (or recovery) problem, all aircrafts and crew need to be recovered to their actual schedule after a disruption. An interesting reference to this problem is Stojković & Soumis (2001). The authors propose a method to recover the aircraft routing and crew schedules simultaneously. Their method is based on a Dantzig-Wolfe decomposition combined with a branch-and-bound method. Another related idea is to make more robust crew schedules where the costs of disruptions are taken into account. Schaefer et al. (2001) discuss some algorithms to take the expected crew costs into account during the optimization and they show that their approach outperformed the traditional one. Finally, Yen & Birge (2000) have used stochastic programming techniques to solve airline crew scheduling problems to get more robust schedules.

The paper is organized as follows. In Section 2, we provide a model and algorithm of the static vehicle and crew scheduling problem, which is used to benchmark the dynamic methods introduced later on in this paper. Before we introduce those approaches, we start the discussion about dynamic vehicle and crew scheduling in Section 3 with some basic assumptions. The dynamic approaches itself are the topic of Section 4. Finally, we conclude this paper with some computational experiments in Section 5.

2 Static Vehicle and Crew Scheduling

In this section we provide a summary of the static approach, which we will use to validate our dynamic approach. We first give a formal problem description of the multiple-depot vehicle and crew scheduling problem. Several approaches to tackle the integrated variant of the vehicle and scheduling problem are recently proposed in the literature (see e.g. Freling (1997), Haase & Friberg (1999), Haase et al. (2001) and Freling et al. (2003) for the single-depot case, and Gaffi & Nonato (1999) and Huisman et al. (2003) for the multiple-depot case). In this paper, we will consider one of the formulations and algorithms proposed by Huisman et al. (2003). For completeness, it is summarized in this section. In fact, the formulation is presented slightly differently here, such that the step to the dynamic problem is smaller.

2.1 Problem Definition

The *multiple-depot vehicle and crew scheduling problem* combines the *multiple-depot vehicle scheduling problem* and the *crew scheduling problem*. Since we will consider dynamic variants of the first two problems later on, we will abbreviate these static problems as S-MDVCSP, S-MDVSP and CSP, respectively.

Given a set of *trips* within a fixed planning horizon, the objective is to minimize the total sum of vehicle and crew costs such that both the vehicle and the crew schedule are feasible and mutually compatible. Each trip has fixed starting and ending times and can be assigned to a vehicle and a crew member from a certain set of depots. Furthermore, the travelling times between all pairs of locations are known. A vehicle schedule is feasible if (1) all trips are assigned to exactly one vehicle, and (2) each trip is assigned to a vehicle from a depot that is allowed to drive this trip. From a vehicle schedule it follows which trips have to be performed by the same vehicle and this defines so-called vehicle *blocks*. The blocks are subdivided at *relief points*, defined by location and time, where and when a change of driver may occur and drivers can enjoy their break. A *task* is defined by two consecutive relief points and represents the minimum portion of work that can be assigned to a crew. These tasks have to be assigned to crew members. The tasks that are assigned to the same crew member define a crew *duty*. Together the duties constitute a crew schedule. Such a schedule is feasible if (1) each task is assigned to one duty, and (2) each duty is a sequence of tasks that can be performed by a single crew, both from a physical and a legal point of view. In particular, each duty must satisfy several complicating constraints corresponding to work load regulations for crews. Typical examples of such constraints are maximum working time without a break, minimum break duration, maximum total working time, and maximum duration. Finally, a *piece (of work)* is defined as a sequence of tasks on one vehicle block without a break that can be performed by a single crew member without interruption.

We distinguish between two types of tasks, viz., *trip tasks* corresponding to trips, and *dh-tasks* corresponding to deadheading. A *deadhead* is a period that a vehicle is moving to or from the depot, or a period between two trips that a vehicle is outside of the depot (possibly moving without passengers).

2.2 Mathematical Formulation

Let $N = \{1, 2, \dots, n\}$ be the set of trips, numbered according to increasing starting time. Define D as the set of depots and let r^d and t^d both represent depot d . Furthermore, define st_i and et_i as respectively the start and ending time of trip i , $trav(r^d, i)$, $trav(i, j)$ and $trav(i, t^d)$ as the deadhead travel time from depot d to the start location of trip i , the deadhead travel time from the end location of trip i to the start location of trip j , and from the end location of trip i to depot d , respectively. Moreover, define $E = \{(i, j) \mid i < j, st_j \geq et_i + trav(i, j)\}$ as the

set of deadheads between trips.

We define the vehicle scheduling network $G^d = (V^d, A^d)$, which is an acyclic directed network with nodes $V^d = N^d \cup \{r^d, t^d\}$, and arcs $A^d = E^d \cup (r^d \times N^d) \cup (N^d \times t^d)$. Note that N^d and E^d are the parts of N and E corresponding to depot d , since it is not necessary that all trips can be served from every depot. Let c_{ij}^d be the variable vehicle cost of arc $(i, j) \in A^d$, which is usually some function of travel and idle time. Furthermore, let c be the fixed cost for using a vehicle.

To reduce the number of constraints, we assume that a vehicle returns to the depot if it has an idle time between two consecutive trips which is long enough to let it return. In that case the arc between the trips is called a *long arc*; the other arcs between trips are called *short arcs*. Denote $A^{d*} \subset A^d$ as the set of arc without long arcs.

Let tp^{dh} , $h = 1, 2, \dots, m$ be the 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. Moreover, define H^d as the corresponding set of timepoints with $tp^{d1} < tp^{d2} < \dots < tp^{dm}$. Let parameter b_i^{dh} be equal to 1 if $st_i \leq tp^{dh} < et_i$, and 0 otherwise. Similarly, let parameters

$$a_{ij}^{dh} = \begin{cases} 1, & \text{if } et_i \leq tp^{dh} < st_j, \\ 0 & \text{otherwise,} \end{cases}$$

for each arc (i, j) with $i, j \in N^d$,

$$a_{r_d j}^{dh} = \begin{cases} 1, & \text{if } st_j - trav(r_d, j) \leq tp^{dh} < st_j, \\ 0 & \text{otherwise,} \end{cases}$$

for each arc (r_d, j) with $j \in N^d$, and

$$a_{i t_d}^{dh} = \begin{cases} 1, & \text{if } et_i \leq tp^{dh} < et_i + trav(i, t_d), \\ 0 & \text{otherwise.} \end{cases}$$

for each arc (i, t_d) with $i \in N^d$.

Furthermore, K^d denotes the set of duties corresponding to depot d and f_k^d denote the crew cost of duty $k \in K^d$, respectively. The subset of duties covering the trip task corresponding to trip $i \in N^d$ is denoted by $K^d(i)$, where we assume that a trip corresponds to exactly one task. $K^d(i, j)$ denotes the set of duties covering dh-tasks corresponding to deadhead $(i, j) \in A^{d*}$. Decision variable y_{ij}^d indicates whether an arc (i, j) is used and assigned to depot d or not, while x_k^d indicates whether duty k corresponding to depot d is selected in the solution or not. Finally, B^d denotes the number of vehicles used from depot $d \in D$. The S-MDVCSP can then be formulated as follows.

$$\min c \sum_{d \in D} B^d + \sum_{d \in D} \sum_{(i,j) \in A^d} c_{ij}^d y_{ij}^d + \sum_{d \in D} \sum_{k \in K^d} f_k^d x_k^d \quad (1)$$

$$\sum_{d \in D} \sum_{\{j:(i,j) \in A^d\}} y_{ij}^d = 1 \quad \forall i \in N, \quad (2)$$

$$\sum_{d \in D} \sum_{\{i:(i,j) \in A^d\}} y_{ij}^d = 1 \quad \forall j \in N, \quad (3)$$

$$\sum_{\{i:(i,j) \in A^d\}} y_{ij}^d - \sum_{\{i:(j,i) \in A^d\}} y_{ji}^d = 0 \quad \forall d \in D, \forall j \in N^d, \quad (4)$$

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

$$\sum_{k \in K^d(i)} x_k^d - \sum_{\{j:(i,j) \in A^d\}} y_{ij}^d = 0 \quad \forall d \in D, \forall i \in N^d, \quad (6)$$

$$\sum_{k \in K^d(i,j)} x_k^d - y_{ij}^d = 0 \quad \forall d \in D, \forall (i,j) \in A^{d*}, \quad (7)$$

$$x_k^d \in \{0, 1\} \quad \forall d \in D, \forall k \in K^d, \quad (8)$$

$$y_{ij}^d \in \{0, 1\} \quad \forall d \in D, \forall (i,j) \in A^d. \quad (9)$$

The objective is to minimize the sum of total vehicle and crew costs. The first four sets of constraints, (2)-(5), correspond to the formulation for the S-MDVSP (see e.g. Huisman et al. (2001)). Constraints (6) assure that each trip task will be covered by a duty from a depot if and only if the corresponding trip is assigned to this depot. Furthermore, constraints (7) guarantee the link between dh-tasks and deadheads in the solution. That is, a vehicle performs deadhead (i, j) if and only if dh-task (i, j) is assigned to a driver from the same depot.

2.3 Algorithm

An outline of the algorithm is shown in Figure 1.

First, we compute a feasible solution by using the sequential approach, which means we compute the optimal solution of the S-MDVSP and afterwards, we solve for each depot a CSP given the vehicle schedule for that depot. To solve the S-MDVSP, we use the model described in Huisman et al. (2001) and the all-purpose solver CPLEX. The approach we used to solve the CSP, is described in Freling et al. (2003).

The main part of the algorithm is used to compute a lower bound and we use therefore a column generation algorithm. The *master problem* is solved with Lagrangian Relaxation. Furthermore, we generate the duties in the column generation subproblem (*pricing problem*). For details about the master and pricing problem, we refer to Huisman et al. (2003). Since we do not want to get a very large master problem, columns with high positive reduced costs will be removed. This only happens if there are more columns than a certain minimum number. Finally, in Step 4 we compute feasible solutions.

In this paper we use the following parameter settings to run the algorithm.

1. The objective is to minimize the total sum of vehicles and drivers. For

Step 0: Initialization

Solve MDVSP and CSP for every depot and take as initial set of columns the duties in the CSP-solution.

Step 1: Computation of dual multipliers

Solve a Lagrangian dual problem with the current set of columns.

This gives a lower bound for the current set of columns.

Step 2: Deletion of columns

If there are more columns than a certain minimum amount, then delete columns with positive reduced cost greater than a certain threshold value.

Step 3: Generation of columns

Generate columns with negative reduced cost.

Compute an estimate of a lower bound for the overall problem.

If the gap between this estimate and the lower bound found in Step 1 is small enough (or another termination criterion is satisfied), go to Step 4; otherwise, return to Step 1.

Step 4: Construction of feasible solution

Solve a second Lagrangian dual problem with the set of columns generated in Step 3, where the optimal solution of the subproblem gives feasible vehicle schedules. Solve for each depot the crew scheduling problem corresponding to the feasible vehicle schedules.

Figure 1: Solution method for S-MDVCSP

solving the S-MDVSP in the sequential approach and in the initial step for the integrated approach we use an additional fictitious cost in the variable vehicle costs, viz., for every minute a vehicle is empty outside the depot a cost equal to 1 is incurred.

2. The pricing problems are solved independently for each depot and each type of duty. Moreover, we generate at most 1500 duties for each combination of a depot and type of duty.
3. The maximum number of iterations in the subgradient algorithm to solve the master problem (Step 1) is $500 + 3k$ in the k -th iteration of the column generation algorithm. However, for constructing the feasible solutions in Step 4, the number of iterations is only 10, since in that case the subproblem is NP-hard. Such a small number of iterations is sufficient, since we already start with good multipliers, namely the best ones of the last iteration in the previous step. We construct 10 feasible solutions from which the best one will be selected.
4. The column generation algorithm is stopped if the difference between the current and estimated lower bound is smaller than 0.1% or if the computation time of the lower bound phase is more than 3 hours. Notice that in

the latter case we do not have a proven lower bound.

3 Assumptions

Before we consider dynamic approaches, we need to realize which aspects play a role in this problem. Several assumptions about these aspects should be made.

Let's consider the following example with two trips, one trip ending at A at 10:00 and another starting from A at 10:15. Furthermore, there should be a break of at least 15 minutes in the duty. Moreover, suppose that both trips are in one duty and the break is between these two trips. However, if the first trip arrives late, we do not have a feasible duty anymore or the second trip starts late. Thus we have the possibility of choosing between violating crew rules and trips starting late. Therefore, we need to make an assumption about the decisions made in such occasions. Throughout the paper, we will assume that passengers have a higher priority than drivers, which means that if there is a possibility to choose between a violation of one or more restrictions of a duty and a trip starting late, the first option is always chosen.

Furthermore, we assume that a trip can only start late due to a delay of the vehicle and thus not due to the driver. That is whenever a changeover occurs this cannot lead to an extra delay on the new vehicle. Since changeovers only occur during the break, we assume that the delays are not larger than the minimum break length. Later on we will see that in our test problems such a delay never occurs and therefore such an assumption is reasonable.

Moreover, we assume that the number of vehicles and crews is unlimited. However, the models and solution approaches can be modified to relax this assumption. As a consequence of this assumption, the number of duties per type are not known beforehand. However, if a duty of a certain specified type has started, the type will not be changed anymore. Therefore, the driver has an indication at which time he will finish his duty, although, he does not know it exactly.

Finally, for evaluation of the schedules we assume that the start time of the two pieces is fixed. Since if a driver reliefs another one at the start of a piece and the bus has a delay, the driver is already there at the scheduled time and does not have the advantage that he could actually start later. Therefore, the total length of pieces, work time and so on, can only increase and not decrease.

4 Different Approaches to Solve the Dynamic Variant

In the case of multiple-depots, all algorithms use the *cluster-reschedule heuristic* shown in Figure 2. This algorithm has an important advantage in practice, since drivers should only be educated about a subset of the trips.

Step 1: Assign the trips to depots by solving (S-MDVSP).
Step 2: For each depot, apply a dynamic approach for the single-depot case.

Figure 2: Solution method for D-MDVCSP

Since multiple-depot problems are solved as several single-depot problems, we only focus on the single-depot case in the remainder of this section. We developed two algorithms to solve the dynamic vehicle and crew scheduling problem with a single depot.

These algorithms use both the following idea: 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 cannot be changed anymore, and we also take into account in some way the future after the next l time units.

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 this approach in which the travel times after the next l periods are simply taken 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. Of course, in principle, it is possible that the planners modify these probabilities or the scenarios themselves during the day. For instance, on a rainy day, they can give higher probabilities to scenarios corresponding to rainy days in the past.

After the $k \leq l$ 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 length 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.

A schematic overview of the first algorithm is provided in Figure 3. This algorithm (called D-VSP-CSP) uses the sequential approach, i.e. first vehicle scheduling and then crew scheduling. The other one (called D-VCSP), which is discussed in Subsection 4.2, uses the integrated approach.

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

Figure 3: Solution method for D-VSP-CSP

Notice that in algorithm D-VSP-CSP we solve problem (D-SDVSP-T), which will be discussed in Subsection 4.1. Afterwards, we solve the corresponding crew schedule for the main scenario, i.e. the scenario with the highest probability (in case there is more than one such scenario, we choose one of them arbitrary). This means that we just have to solve a standard crew scheduling problem. However, we need to take into account that decisions made before time point T cannot be changed anymore. This can be handled in the construction of the set of duties K . In fact the requirement that there should be no contradiction with decision made earlier, is just an extra feasibility check when we construct this set. Therefore, this does not have any impact on the structure of the underlying set partitioning problem.

4.1 Mathematical Formulation (D-SDVSP-T)

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 , $trav(r, i)$ and $trav(i, t)$ 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 $trav(i, j)$ 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 $st_j - trav(r, j) \in [T, T + l)$;
- (i, j) , if $st_j - trav(i, j) \in [T, T + l)$;
- (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_{ij}^s as the cost of arc (i, j) in respectively period $[T, T + l)$ and the period after $T + l$ in scenario s . Here, the costs associated with an arc is 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.

Similar as in Section 2.2, 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

$$a_{ij}^{sh} = \begin{cases} 1, & \text{if } et_i^s \leq tp^{sh} < st_j, \\ -1, & \text{if } st_j \leq tp^{sh} < et_i^s, \\ 0 & \text{otherwise,} \end{cases}$$

for each arc (i, j) with $i, j \in N$,

$$a_{rj}^{sh} = \begin{cases} 1, & \text{if } st_j - trav(r, j) \leq tp^{sh} < st_j, \\ 0 & \text{otherwise,} \end{cases}$$

for each arc (r, j) with $j \in N$, and

$$a_{it}^{sh} = \begin{cases} 1, & \text{if } et_i^s \leq tp^{sh} < et_i^s + trav(i, t), \\ 0 & \text{otherwise,} \end{cases}$$

for each arc (i, t) with $i \in N$.

For arcs $(i, j) \in A_1$ similar definitions hold, but then with et_i^0 instead of et_i^s . Furthermore, let b^{sh} be the number of trips carried out at tp^{sh} , i.e., we count all trips for which $st_i \leq tp^{sh} < et_i^0$ for trips ending in $[T, T + l)$ and $st_i \leq tp^{sh} < et_i^s$ 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, since 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 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):

$$\min c \sum_{s \in S} p^s B^s + \sum_{(i,j) \in A_1} c'_{ij} z_{ij} + \sum_{s \in S} p^s \sum_{(i,j) \in A_2} c^s_{ij} y^s_{ij} \quad (10)$$

$$\sum_{\{i:(i,j) \in A_1\}} z_{ij} + \sum_{\{i:(i,j) \in A_2\}} y^s_{ij} = 1 \quad \forall s \in S, \forall j \in N, \quad (11)$$

$$\sum_{\{j:(i,j) \in A_1\}} z_{ij} + \sum_{\{j:(i,j) \in A_2\}} y^s_{ij} = 1 \quad \forall s \in S, \forall i \in N, \quad (12)$$

$$b^{sh} + \sum_{(i,j) \in A_1} a^{sh}_{ij} z_{ij} + \sum_{(i,j) \in A_2} a^{sh}_{ij} y^s_{ij} \leq B^s \quad \forall s \in S, \forall h \in H^s, \quad (13)$$

$$z_{ij} \in \{0, 1\} \quad \forall (i, j) \in A_1, \quad (14)$$

$$y^s_{ij} \in \{0, 1\} \quad \forall s \in S, \forall (i, j) \in A_2. \quad (15)$$

Constraints (11) and (12) assure that every trip has exactly one predecessor and one successor in every scenario. Furthermore, constraint (11) 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 (12) and the predecessor of a trip. Finally, constraint (13) and the fact that $c > 0$ guarantee that B^s is the number of vehicles in scenario s .

4.2 Algorithm D-VCSP

An overview of algorithm D-VCSP is given in Figure 4.

<p>Step 0: Choose initial parameters T, k and l ($\geq k$).</p> <p>Step 1: Solve problem (D-SDVCSP-T) for period $[T, T + l)$. Update T: $T := T + k$. Repeat step 1 until the end of the day.</p>

Figure 4: Solution method for D-VCSP

The algorithm solves a sequence of integrated vehicle and crew scheduling problems. In this subsection we provide a mathematical formulation for this problem. Hereby, we assume again that one of the scenarios (s^*) is the main scenario and that the variables related to the crew scheduling part of the problem are defined on this scenario. We only consider the general case with a single depot.

4.2.1 Mathematical Formulation (D-SDVCSP-T)

Before providing the mathematical formulation we need to recall some notation. For the notation with respect to the vehicle scheduling part of the formulation we refer to Subsection 4.1, since it is completely similar.

Furthermore, similar as in Section 2.2, K denotes the set of duties, f_k denotes the crew cost of duty $k \in K$ and $K(i, j)$ denotes the set of duties covering dh-tasks corresponding to deadhead $(i, j) \in A^*$. Finally, binary decision variables x_k are defined as 1 if duty k is selected in the solution and 0 otherwise.

The problem can be formulated as follows.

(D-SDVCSP-T):

$$\min c \sum_{s \in S} p^s B^s + \sum_{(i,j) \in A_1} c'_{ij} z_{ij} + \sum_{s \in S} p^s \sum_{(i,j) \in A_2} c^s_{ij} y^s_{ij} + \sum_{k \in K} f_k x_k \quad (16)$$

$$\sum_{\{i:(i,j) \in A_1\}} z_{ij} + \sum_{\{i:(i,j) \in A_2\}} y^s_{ij} = 1 \quad \forall s \in S, \forall j \in N, \quad (17)$$

$$\sum_{\{j:(i,j) \in A_1\}} z_{ij} + \sum_{\{j:(i,j) \in A_2\}} y^s_{ij} = 1 \quad \forall s \in S, \forall i \in N, \quad (18)$$

$$b^{sh} + \sum_{(i,j) \in A_1} a^{sh}_{ij} z_{ij} + \sum_{(i,j) \in A_2} a^{sh}_{ij} y^s_{ij} \leq B^s \quad \forall s \in S, \forall h \in H^s, \quad (19)$$

$$\sum_{k \in K(i)} x_k = 1 \quad \forall i \in N, \quad (20)$$

$$\sum_{k \in K(i,j)} x_k - z_{ij} = 0 \quad \forall (i, j) \in A_1, \quad (21)$$

$$\sum_{k \in K(i,j)} x_k - y^{s*}_{ij} = 0 \quad \forall (i, j) \in A_2, \quad (22)$$

$$x_k, z_{ij} \in \{0, 1\} \quad \forall k \in K, \forall (i, j) \in A_1, \quad (23)$$

$$y^s_{ij} \in \{0, 1\} \quad \forall (i, j) \in A_2, \forall s \in S. \quad (24)$$

The objective function (16) minimizes the total sum of vehicle, crew and delay costs. The first three sets of constraints, (17) - (19), correspond to the formulation of the dynamic vehicle scheduling problem (see Subsection 4.1). Constraints (20) assure that each trip task is in a duty. Finally, the constraints (21) and (22) guarantee the link between dh-tasks and deadheads in the solution. Notice that it is not necessary here to distinguish between short and long arcs since the fixed vehicle costs are dealt with separately.

4.2.2 Algorithm (D-SDVCSP-T)

The algorithm for D-SDVCSP-T is shown in Figure 5.

The differences between this algorithm and the one for VCSP1 are in the steps 0, 1 and 3. Step 2 is exactly similar, since it only deals with generation of columns and another vehicle scheduling problem does not influence this. The main difference is found in Step 1, where constraints (20)-(22) are first replaced by set covering constraints, which are subsequently relaxed in a Lagrangian way. That is, we associate non-negative Lagrangian multipliers λ_p , μ_{ij} and ν^s_{ij} with

Step 0: Initialization

Solve D-SDVSP-T and CSP for the main scenario and take as initial set of columns the duties in the CSP-solution.

Step 1: Computation of dual multipliers

Solve a Lagrangian dual problem with the current set of columns.

This gives a lower bound for the current set of columns.

Step 2: Generation of columns

Generate columns (duties) with negative reduced cost.

Compute an estimate of a lower bound for the overall problem.

If the gap between this estimate and the lower bound found in Step 1 is small enough (or another termination criterion is satisfied), go to Step 3; otherwise, return to Step 1.

Step 3: Construction of feasible solution

Based on feasible vehicle solution(s) for scenario s^* from Step 1, construct corresponding feasible crew solution(s).

Figure 5: Solution method for D-SDVCSP-T

constraints (20), (21) and (22), respectively. Then the remaining Lagrangian subproblem can be solved by pricing out the x variables and the following problem for the y and z variables:

$$\min c \sum_{s \in S} p^s B^s + \sum_{(i,j) \in A_1} (c'_{ij} - \mu_{ij}) z_{ij} + \sum_{s \in S} \sum_{(i,j) \in A_2} \bar{c}_{ij}^s y_{ij}^s \quad (25)$$

$$\sum_{\{i:(i,j) \in A_1\}} z_{ij} + \sum_{\{i:(i,j) \in A_2\}} y_{ij}^s = 1 \quad \forall s \in S, \forall j \in N, \quad (26)$$

$$\sum_{\{j:(i,j) \in A_1\}} z_{ij} + \sum_{\{j:(i,j) \in A_2\}} y_{ij}^s = 1 \quad \forall s \in S, \forall i \in N, \quad (27)$$

$$b^{sh} + \sum_{(i,j) \in A_1} a_{ij}^{sh} z_{ij} + \sum_{(i,j) \in A_2} a_{ij}^{sh} y_{ij}^s \leq B^s \quad \forall s \in S, \forall h \in H^s, \quad (28)$$

$$z_{ij} \in \{0, 1\} \quad \forall (i, j) \in A_1, \quad (29)$$

$$y_{ij}^s \in \{0, 1\} \quad \forall s \in S, \forall (i, j) \in A_2, \quad (30)$$

where

$$\bar{c}_{ij}^s = \begin{cases} p^s c_{ij}^s - \nu_{ij}^s & \text{if } (i, j) \in A_2, s = s^*, \\ p^s c_{ij}^s & \text{if } (i, j) \in A_2, s \neq s^*. \end{cases}$$

Notice that this problem is equivalent to problem (D-SDVSP-T). Moreover, we will again use the CPLEX MIP solver to compute an optimal solution for this problem.

At the end we compute a feasible crew schedule given the (feasible) vehicle schedule for scenario s^* which resulted from solving the last Lagrangian subproblem. Of course, it is possible to compute more feasible solutions by solving the

CSP not only for the vehicle solution from the last iteration, but also for vehicle solutions which were encountered earlier on.

5 Computational Experience

We have evaluated our approach by using a small single-depot and a medium-sized multiple-depot data set from Connexxion. The problems, denoted as `prob_A` and `prob_B`, consist of 164 and 304 trips, respectively. The restrictions w.r.t. the feasibility of a duty that have to be taken into account are described in Subsection 5.1. Furthermore, we discuss there the assumptions about the realizations of the travel times. Finally, notice that all tests reported in this subsection are executed on a Pentium III 450MHz personal computer (128MB RAM).

We consider in this section six measurements to evaluate a solution: the total number of vehicles and drivers used, the percentage of trips starting late, the “virtual” delay costs, the percentage of duties violating at least one of the crew rules and a “virtual” costs measure for those violations.

5.1 Data Description

In this subsection, we discuss some important characteristics of the data sets. The set consists of trips and 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). 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 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 is 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, since 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.

The restrictions w.r.t. the duties that we have taken into account, are as follows. A driver can only be relieved by another driver at the start or end of a trip at certain specified locations or at the depot. There are 8 of these locations, which are all major bus stations. If a driver starts/ends his duty at the depot, there is a sign-on/sign-off time of 10 and 5 minutes, respectively. If a driver starts/ends his duty at another relief location, an extra time of 15 minutes plus the deadhead time between this location and the depot is added to the length of the duty. There are five different types of duties, one tripper type consisting of one piece with a length between 30 minutes and 5 hours, and four normal types consisting of two pieces with the properties described in Table 1.

type	1 (early)		2 (day)		3 (late)		4 (split)	
	min	max	min	max	min	max	min	max
start time			8:00		13:15			
end time		16:30		18:14				19:30
piece length	0:30	5:00	0:30	5:00	0:30	5:00	0:30	5:00
break length	0:45		0:45		0:45		1:30	
duty length		9:45		9:45		9:45		12:00
work time		9:00		9:00		9:00		9:00

Table 1: Properties of the different duty types

Moreover, the maximum delay that occurs in these data is less than 45 minutes, the minimum break length. Therefore, the second assumption made in Section 3, i.e. the one that a trip can only start late due to a delay of the vehicle, holds.

5.2 Results Static Approach

We use 1,000 as fixed costs per vehicle and variable vehicle costs of 1 per minute time that a vehicle is without passengers. Furthermore, we use 1,000 as fixed costs for each duty. To solve the static problem, we can choose between the sequential and the integrated approach. In the sequential approach we solve the static vehicle scheduling problem and afterwards the crew scheduling problem, while in the integrated approach we solve the whole problem at once. We will use the same parameter settings as described in Subsection 2.3.

Furthermore, we calculate, for the 10 days of which we have data of the realizations of the travel times, the percentage of trips that start late and the cost of these delays for which we use as cost function $10x^2$, where x is the time in minutes that the trip starts late. Finally, we check, for these 10 days, the percentage of duties that violate one or more of the restrictions and the cost of these violations. Here, we use for each duty the cost function $10y_1^2 + 2y_2^2 + 2y_3^2 + y_4^2 + y_5^2 + y_6^2$, where y_1 is the number of minutes that the break in the duty is shorter than the minimum break length, y_2 (y_3) is the number of minutes that the maximum duty length (working time) is exceeded, y_4 (y_5) is the number of minutes that the maximum length of the first (second) piece is exceeded and y_6 is the number of minutes that the latest end time is exceeded. We calculate the total violation costs as the sum of these costs of each duty.

In Tables 2 and 3, the best feasible solutions for prob_A and prob_B of the static as well as the integrated approach are denoted, respectively. Furthermore, we give the following averages, which are explained above, over the 10 days: the percentage of trips starting late, the delay costs, the percentage of duties violated and the violation costs. We give these results without buffer times (standard problem) and with buffer times of 2, 5, 7 and 10 minutes, respectively. In this table #V and #D denote the number of vehicles and drivers used, respectively, %L the percentage of trips starting late, DC the cost of these delays, %DV the percentage of duties violated and VC the total cost of these violations.

buffer	no		2 min.		5 min.		7 min.		10 min.	
	seq.	int.	seq.	int.	seq.	int.	seq.	int.	seq.	int.
#V	17	17	20	20	24	24	24	24	25	25
#D	32	31	35	32	40	37	41	39	42	41
%L	16.2	15.1	8.8	7.6	3.9	3.8	2.4	3.0	2.0	2.2
DC	17,272	17,812	7,765	7,360	2,925	3,927	2,612	3,264	2,681	2,178
%DV	7.8	11.6	7.4	10.0	3.8	4.9	4.1	4.6	6.2	5.6
VC	825	569	538	1,224	934	69	97	164	322	191

Table 2: Average results of the S-VCSP - prob_A

It is obvious that the number of vehicles and drivers increases when we introduce buffer times, while the percentage of trips starting late (and its costs) decreases. Furthermore, it is noteworthy to mention that the effect of buffer times is much higher in prob_A than in prob_B. The numbers of vehicles and drivers increase with about 50% and 30%, respectively, by introducing a buffer time of 10 minutes in prob_A, while this is only about 10% in prob_B. This effect is caused by several differences in the data. First of all, prob_B has more trips

buffer	no		2 min.		5 min.		7 min.		10 min.	
	seq.	int.	seq.	int.	seq.	int.	seq.	int.	seq.	int.
#V	40	40	41	41	43	43	44	44	44	44
#D	74	65	76	66	81	71	82	71	82	71
%L	9.9	9.2	5.6	4.9	2.9	2.3	2.1	1.9	1.8	1.5
DC	19,897	15,691	8,742	7,206	3,906	4,830	2,741	2,721	1,963	1,994
%DV	8.9	15.7	5.8	10.7	7.6	10.3	4.2	9.4	8.5	8.4
VC	1,499	6,935	1,153	6,865	3,612	2,211	1,621	5,320	4,495	5,080

Table 3: Average results of the S-VCSP - prob_B

and therefore, there are more options to combine trips. Secondly, the duration of the trips in prob_A is slightly smaller, which means that the same buffer time is relatively larger for prob_A.

5.3 Results Dynamic Approach

We show the results of our approach to the dynamic vehicle and crew scheduling problem and compare them 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 . We only consider the case where l is equal to 120 and 60 minutes.
- Rescheduling only takes place at the start of a period ($k = l$).
- With respect to the vehicle and crew costs, the cost structure is the same as in the previous subsection.
- A cost of 500 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. There are no costs taken into account for violating duties during the optimization. However, we still use the cost function from the previous subsection to evaluate the solutions.
- 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 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 II), we computed this average scenario by the weighted average of the 9 scenarios as described above.
- Mostly, we solve the integrated problems (in case I as well as case II) with the same parameter settings as in the static case (see Subsection 2.3). However, sometimes we deviate from those settings. We will explicitly mention those deviations.

We applied the four heuristics described in Section 4 to the dynamic vehicle and crew scheduling. In Tables 4 and 5, the average results over all days are shown for prob_A and prob_B, respectively, for the cases I and II using the sequential algorithm (D-VSP-CSP) as well as the integrated one (D-VCSP). Hereby, the lower bound phase of the integrated approach is stopped after 11 iterations such that the computation time is reasonable low, which is necessary in a dynamic environment. Moreover, we use as period length, l , 120 minutes. In this table #V, #D, %L, DC, %DV and VC have the same meaning as before.

	I		II	
	sequential	integrated	sequential	integrated
#V	18.7	18.7	18.6	18.6
#D	43.9	37.8	40.2	37.2
%L	3.4	2.9	3.2	3.8
DC	2,986	1,798	2,814	3,351
%DV	6.1	5.6	4.5	5.4
VC	445	316	361	637

Table 4: Average results D-VCSP - prob_A

If we compare the different approaches the most important criteria are, of course, the actual costs, i.e. the number of vehicles and crews. If these results are similar, we take the other criteria into account. Therefore, we can immediately conclude that the integrated approach significantly outperforms the sequential approach. This is also what we expected beforehand, since applying an integrated approach leads in each iteration to an improvement of the sequential solution and

	I		II	
	sequential	integrated	sequential	integrated
#V	41	40.4	41.1	40.6
#D	85.6	76.5	84.9	76.4
%L	2.4	4.2	2.8	4.2
DC	3,509	6,586	4,173	7,633
%DV	3.9	4.8	5.2	5.3
VC	598	1,201	1,567	633

Table 5: Average results D-VCSP - prob_B

therefore, the overall improvement is dramatic. Furthermore, we can see that the difference between case I and II (in the sequential as well as the integrated approach) is small. However, introducing more scenarios (case I) leads to a slightly better solution. Finally, we have to conclude that for prob_B even the best dynamic approach performs worse than the static integrated approach with fixed buffer times. Possible reasons for this are the stopping criteria in the dynamic version of the integrated approach and the way the problem is split up. We will investigate this question later on, but we will now focus on the results of the integrated approach (case I) for prob_A. In Table 6, we provide some more results for other values of the period length, while we give the total computation time (denoted by CPU), the number of iterations (it.) and the maximum computation time of one iteration over all instances (max.) in Table 7.

l	#V	#D	%L	DC	%DV	VC
120	18.6	37.2	3.8	3,351	5.4	637
60	18.3	36.4	5.1	4,272	5.7	568

Table 6: Detailed results D-VCSP - case I integrated - prob_A

l	CPU	it.	max.
120	13,578	10	5,612
60	26,370	19	17,336

Table 7: Computation times D-VCSP - case I integrated - prob_A

We can see that the results are better than in the static case with buffer times. Consider for example the case with 5 minutes buffer time, where we needed 24 vehicles and 37 drivers. In the dynamic approach, with l equal to 120, we can save more than 5 vehicles, while the number of drivers only slightly increases, the percentage of trips starting late is the same, and the percentage of violated duties and its costs are only slightly higher. Moreover, the delay costs are even slightly lower. Of course, we have assumed here that we know all travel times two hours ahead, in fact two hours before we start the calculation which also takes about 1,5 hour in the worst case. Therefore, the travel times are assumed to be known 3,5 hours before their realization. Due to this large computation time we cannot consider too small values of l . Furthermore, we can see that if the period length is 60 minutes, the results only change slightly. A few less vehicles and drivers are needed and there are slightly more delays. However, the maximum computation of one iteration is much larger than l , which means that we cannot use it anymore in practice. We can even see that this time is much larger than in the case where $l = 120$. This is a coincidence since the average computation of an iteration is almost equal in both cases (1,358 compared to 1,388 seconds).

To explain the bad performance of the dynamic approaches for prob_B, we also considered a variant of the D-VCSP without a maximum number of iterations (or indirectly computation time) in the lower bound phase of the integrated approach. That is the lower bound phase is only terminated when no duties with negative reduced costs have been found or the difference between the lower bound on the overall set of duties and the lower bound in a certain iteration is very small (less than 0.1%). For this purpose we only considered case I. Furthermore, we looked at the results for smaller values of l , the length of the period. The average results over all days are denoted in Table 8. The computation times for these cases are shown in Table 9. Notice hereby that the problem has been split into one subproblem for each depot. The abbreviations in both tables have the same meaning as before.

l	#V	#D	%L	DC	%DV	VC
120	40.5	75.4	4.4	7,101	6.4	1,761
60	40.5	76.5	5.0	8,809	5.1	1,692

Table 8: Improved results D-VCSP - case I integrated

We can see that the effect of allowing more computation time is quite small. On average, one driver can be saved, but on the other criteria the performance is slightly worse. The maximum computation time per iteration and per depot is slightly less than one hour, which is only reasonable for large values of l . Finally, we have to conclude that if we also consider crew scheduling aspects, the dynamic approach is not as good as we expected beforehand. It does not perform well

	depot 1			depot 2			depot 3			depot 4		
l	CPU	it.	max.	CPU	it.	max.	CPU	it.	max.	CPU	it.	max.
120	1,861	8	513	2,378	10	364	10,374	10	2,690	240	7	48
60	1,949	14	335	2,763	19	267	16,093	19	2,716	410	13	48

Table 9: Computation times D-VCSP - case I integrated

under all circumstances.

5.3.1 Lower Bound

Since the results of the dynamic approach are not completely satisfactory, we consider the situation where we have perfect information and compare several variants such that we can see where things go wrong. In the case with perfect information, there are obviously no trips starting late and thus also no delay costs and violated duties. Therefore, we just calculate the best feasible solution for each day with as input the realization of the travel times of that particular day. We do this again with the sequential and integrated approach. Notice that for such large problems no lower bounds on the optimal (integrated) solution can be found in a reasonable time. Therefore, we do not mention them here. The average of the best feasible solutions over all days can be found in Tables 10 and 11 for prob_A and prob_B, respectively.

	sequential	integrated
#V	19.4	19.4
#D	33.4	32.3

Table 10: Results in the case of perfect information - prob_A

	sequential	integrated
#V	40.8	40.8
#D	75.5	66.9

Table 11: Results in the case of perfect information - prob_B

From these tables, we can conclude that the number of vehicles and drivers is of course slightly higher than in the static case without buffer times. However, the difference is very small. A remaining question is now whether the poor quality

of the dynamic approach for prob_B can be explained by the fact that we use a cluster-reschedule heuristic, i.e. we first split the problem into several smaller problems for each depot, or by the performance of the heuristics, which we used to solve a single-depot problem. This question can be partly answered by looking at the results in the case of perfect information by using the same splitting of the problem into several subproblems. These average results over all days are denoted in Table 12 for the sequential as well as the integrated approach. We can see that especially the difference for the integrated approach is quite large, namely a total cost saving of 7.8%. Therefore, we can conclude that the division of the problem already leads to a significant loss in the quality of the solution. An interesting question is thus how to divide large problem instances into several smaller ones such that an overall good solution will be obtained. This question has been discussed in Huisman (2004) and will be subject of further research.

	sequential	integrated
#V	43.3	43.3
#D	77.7	73.5

Table 12: Results in the case of perfect information by first splitting the problem

5.3.2 Sensitivity Analysis

The assumption that the travel times are known (can be estimated without any error) at time point T for the period $[T, T + l)$ may be unrealistic for the considered values of l . Therefore, we have performed a sensitivity analysis by considering small deviations of the actual travel times from the estimated ones. We have only done this analysis for prob_A, since the dynamic approach did not perform well for prob_B even if the travel times were exactly known. Notice that by using slightly different simulated travel times than the estimated ones, only the performance w.r.t the number of trips starting late, the delay costs, the number of violated duties and the violation costs can change. The number of vehicles and drivers do not change.

In the same way as in Huisman et al. (2001), 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. The results can be found in Table 13.

As can be easily seen from the table, the effects of small deviations from the estimated travel times are small. In fact, the effects of disturbances are much smaller than in Huisman et al. (2001), where we only considered vehicles. This means that a dynamic approach can be used to solve prob_A.

l	σ	%L	DC	%DV	VC
120	2	11.1	4,534	6.4	687
120	1	6.9	3,652	5.7	652
120	0.5	5.0	3,467	5.5	646
120	0.25	4.0	3,394	5.4	644
60	2	10.7	5,434	6.7	609
60	1	7.7	4,740	6.4	582
60	0.5	6.6	4,475	6.3	580
60	0.25	6.6	4,417	6.2	591

Table 13: Average results with extra perturbation $\epsilon \sim N(0, \sigma^2)$ - prob_A

5.4 Discussion and Conclusion

In this paper we considered the dynamic vehicle and crew scheduling. The extension of our approach for the dynamic vehicle scheduling problem to the situation where we also considered crews did not always give the results which we expected beforehand. For the small instance with a single depot the dynamic approach performs well. However, computation times may be too high to apply such an approach in practice. On the other hand for the medium-sized instance with multiple depots, the traditional static approach with buffer times performed much better. We discussed several reasons why our dynamic approach does not perform so well in this case. The first reason is that this approach used the cluster-reschedule heuristic, i.e. all trips are assigned beforehand to a certain depot. In other words, the overall result is dependent on the chosen assignment. Another reason could be that the computation times allowed to solve the approach dynamically were set too small, although by extending these times the results did not improve so much. The final mentioned reason is that the idea of dynamically solving itself does not work so well. Since the dynamic approach worked well for the small instance where the data set was not divided into several smaller ones, and since extending the computation times did not lead to significant improvement, we can conclude that the way the problem is split up, is the bottleneck. Therefore, we recommend to invest further research in speeding up the suggested algorithms. With faster computers and better algorithms the dynamic approach should outperform the static one with buffer times for larger problem instances as well.

Finally, we would like to make several remarks about the practical applicability of such a dynamic approach. First of all, it will be difficult to test the assumptions we made in a practical environment. For instance, how can one mea-

sure if the travel times of the trips are really independent of the actual chosen schedule? Secondly, it is important how drivers (but also planners and managers) react on such a way of working. It is very easy for them to frustrate such an approach. Therefore, we conclude this discussion with the fact that there is still a long way to go before such an approach can be used in practice!

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

References

- Freling, R. 1997. Models and Techniques for Integrating Vehicle and Crew Scheduling, Ph.D. thesis, Tinbergen Institute, Erasmus University Rotterdam.
- Freling, R., Huisman, D., Wagelmans, A.P.M., 2003. Models and Algorithms for Integration of Vehicle and Crew Scheduling, *Journal of Scheduling* 6, 63–85.
- Gaffi, A., Nonato, M., 1999. An Integrated Approach to Extra-Urban Crew and Vehicle Scheduling. In: Wilson, N.H.M. (Ed.), *Computer-Aided Transit Scheduling*, Springer Verlag, Berlin, pp. 103–128.
- Haase, K., Friberg, C., 1999. An Exact Branch and Cut Algorithm for the Vehicle and Crew Scheduling Problem. In: Wilson, N.H.M. (Ed.), *Computer-Aided Transit Scheduling*, Springer Verlag, Berlin, pp. 63–80.
- Haase, K., Desaulniers, G., Desrosiers, J., 2001. Simultaneous Vehicle and Crew Scheduling in Urban Mass Transit Systems, *Transportation Science* 35, 286–303.
- Horner, P., 2002. How Continental Landed on its Fleet, *OR/MS Today* 29, 30–31.
- Huisman, D., Freling, R., Wagelmans, A.P.M., 2001. A Dynamic Approach to Vehicle Scheduling, Tech. rept. EI2001-17, Econometric Institute, Erasmus University Rotterdam, Rotterdam. To appear in *Transportation Science* under the title “A Robust Solution Approach to the Dynamic Vehicle Scheduling Problem”.
- Huisman, D., Freling, R., Wagelmans, A.P.M., 2003. Multiple-Depot Integrated Vehicle and Crew Scheduling, Tech. rept. EI2003-02, Econometric Institute, Erasmus University Rotterdam, Rotterdam. To appear in *Transportation Science*.

- Huisman, D. 2004. Integrated and Dynamic Vehicle and Crew Scheduling, Ph.D. thesis, Tinbergen Institute, Erasmus University Rotterdam.
- Schaefer, A.J., Johnson, E.L., Kleywegt, A.J., Nemhauser, G.L., 2001. Airline Crew Scheduling under Uncertainty, Tech. rept. Georgia Institute of Technology, Atlanta, Georgia.
- Stojković, M., Soumis, F., 2001. An Optimization Model for the Simultaneous Operational Flight and Pilot Scheduling Problem, *Management Science* 47, 1290–1305.
- Yen, J.W., Birge, J.R., 2000. A Stochastic Programming Approach to the Airline Crew Scheduling Problem, Tech. rept. (Downloadable from website <http://faculty.washington.edu/joyceyen>).