Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Procedia Social and Behavioral Sciences 20 (2011) 292–301
Procedia
Social and Behavioral Sciences

14th EWGT & 26th MEC & 1st RH

Increasing delay-tolerance of vehicle and crew schedules in public transport by sequential, partial-integrated and integrated approaches

Bastian Amberg[a,*], Boris Amberg[b], Natalia Kliewer[a]

[a]*Freie Universität Berlin, Garystr. 21, 14195 Berlin, Germany*
[b]*Decision Support & OR Lab and International Graduate School of Dynamic Intelligent Systems, University of Paderborn, Warburger Str. 100, 33098 Paderborn, Germany*

Abstract

In public bus transport delays occur frequently during execution of vehicle and crew schedules. Delayed busses not only affect the vehicle schedule but also the associated crew schedule. Delayed drivers cause similar effects the other way round. Thus planned schedules can become infeasible and the operations control has to initiate expensive recovery actions. To avoid these undesirable effects possible disruptions can already be considered in the planning phase. We present different offline approaches to increase delay-tolerance of both vehicle schedules and crew schedules. Thereby we consider sequential, partial-integrated and integrated planning methods for vehicle and crew scheduling.
We propose two different types of approaches: approaches using no information about possible delays, and approaches using historical delay data. Within these approaches the main focus is on providing buffer times at appropriate positions in the vehicle and crew schedules. Buffer times should be distributed that way that minor disruptions are absorbed and delay propagation can be limited. Further, planned costs should not be increased significantly compared to cost-optimal scheduling. We use different flow decomposition strategies for cost-optimal flows of the underlying time-space-network model in order to redistribute buffer times. In addition, we apply different strategies to select timetabled trips that might be susceptible to delays. After these trips a certain minimum buffer time is guaranteed. The approaches are compared with regard to planned costs and delay-tolerance using real-life timetables from German cities for the experiments.

Keywords: vehicle and crew scheduling; public transport; delay-tolerance; robustness; time-space-network

1. Introduction

In order to support the generation of cost-efficient vehicle and driver schedules, public transport companies increasingly use software tools, containing state-of-the-art optimization methods. In this work we examine scheduling approaches that increase the delay-tolerance of vehicle and crew schedules in public bus transport without increasing planned costs too much compared to cost-optimal scheduling. However, using optimization

* Corresponding author. Tel.: +49-30-838-53533; fax: +49-30-838-52027.
E-mail address: bastian.amberg@fu-berlin.de

doi:10.1016/j.sbspro.2011.08.035

methods to compute cost-optimal schedules usually results in tense schedules for both vehicles and crews without much idle or waiting time. Thus, during execution of vehicle and crew schedules delays cannot be absorbed and can propagate through the whole schedule. As a result, planned schedules can become infeasible and the operations control has to initiate expensive recovery actions.

In public bus transport delays occur frequently during execution of vehicle and crew schedules. Delays can be caused by traffic, increased number of passengers, road work, accidents, bus breakdown and so on. In the following we give some exemplary data according to the frequency of delays during the daily operations of public transport companies. The statistics have been published by two German transport companies. First data is taken from the public transport company operating in Berlin. This company serves about 149 lines with a fleet of 1,349 buses. In 2009, only 86.6 % of all serviced trips were operated with a delay less than 3 minutes. Second example is published by a transport company operating in the area of Nuremberg. This company serves 70 lines with a fleet of 282 buses. In 2009, only 86% of all service trips were operated with a delay less than 2 minutes.

Delays are unavoidable during operations but schedules can be created that way that possible disruptions can be absorbed. However, disruptions can already be considered in the planning phase. We present different offline approaches that increase delay-tolerance of both vehicle schedules and crew schedules by redistributing buffer times within the schedules or by inserting fixed or various buffer times. Thereby we consider sequential, partial-integrated and integrated planning methods for vehicle and crew scheduling. To the best of our knowledge, such an investigation about increasing delay-tolerance within both vehicle and crew schedules in public bus transport has not been previously published.

The paper is organized as follows: A detailed problem description is given in section 2. Section 3 briefly describes the state-of-the-art approaches for vehicle and crew scheduling. Section 4 gives an overview about our solution framework. In section 5 and in section 6 we propose approaches to increase delay-tolerance without increasing planned cost compared to cost-optimal scheduling. Section 5 is about redistributing buffer times based on the vehicle scheduling solution, whereas in section 6 methods are proposed to redistribute buffer times based on the crew scheduling solution. Section 7 briefly describes strategies about adding buffer time before connecting timetabled trips. In section 8 we present computational results comparing different variants of our approaches. Finally, in section 9 conclusions and outlook to future research are drawn.

2. Problem definition

For a given set of timetabled trips the *vehicle scheduling problem* (VSP) can be stated as follows: Find an assignment of trips to vehicles, such that each trip is assigned exactly once, each vehicle performs a feasible sequence of trips, each sequence starts and ends at the same depot, constraints on the number and on the types of vehicles are considered, and the fleet size and operational cost are minimized. A daily schedule for one vehicle (also called *vehicle duty*) consists of several *vehicle blocks*. Vehicle blocks are sequences of served trips between two layovers at a depot. After serving one timetabled trip, also called *service trip*, each bus can serve one of the trips starting later from its arrival station, or it can change its location by moving empty to another station by an *empty movement*, or *deadhead,* in order to serve a service trip starting there. Deadheads from and to a depot are called *pull-out* and *pull-in* trips, respectively.

The *crew scheduling problem* (CSP) is similar to the vehicle scheduling problem; it additionally contains constraints concerning work regulations for crews. The problem deals with assigning *tasks* to (day) duties such that each task is executed, each duty is feasible, and the total costs of the duties are minimized. The basic elements of duties (tasks) correspond to sequences of activities (e.g. scheduled trips or deadheads) between consecutive relief points and represent an elementary portion of work that can be assigned to a driver. A *relief point* defines a location and time where a driver may change his vehicle. A sequence of tasks for which a driver stays in the same vehicle is called *piece of work*. Pieces of work separated by breaks form a *duty*. Typically there are several duty types in practical applications, each with a different rule set. Examples of working rules are a minimum/maximum working and driving time, minimum break length, or maximum duration of a duty.

Vehicle and crew scheduling are usually approached in a sequential manner, i.e. vehicle schedules are determined before crew schedules. However, integrating vehicle and crew scheduling to solving both problems simultaneously, basically enables further cost-saving opportunities because the degrees of freedom and consequently the size of the solution space are increased.

The *integrated vehicle and crew scheduling problem* (IVCSP) includes VSP and CSP as subproblems: For a given set of timetabled trips, depots, and relief points minimum cost sets of vehicle blocks and crew duties have to be found, such that vehicle and crew schedule are feasible and mutually *compatible*. Vehicle and crew schedules are compatible if each trip of the timetable is covered by a vehicle and a duty, and each deadhead used in the vehicle schedule is also covered by exactly one duty.

Schedules are called *robust* when effects of disruptions are less likely to be propagated into the future. There are different ways to measure robustness. Within this work we concentrate on the *delay-tolerance* of a schedule and use the measure proposed by Kramkowski et al. (2009) in the context of pure vehicle scheduling. Therefore, we differentiate between *primary* and *secondary* delays. Primary delays are exogenous and are directly caused by disruptions. In contrast, secondary delays are endogenous and can be influenced by modification of vehicle and/or crew schedule. They are induced by primary delays depending on the structure of the schedules. Figure 1 contains an example about primary and secondary delays and their dependency. Due to a disruption the planned arrival time (pAT) of trip 1 cannot be maintained. Thus, the planned departure time (pDT) of the following trip 2 cannot be met, too. The difference between planned departure time and actual departure time (DT) of trip 2 is called secondary (or propagated) delay. Notice, that in this work we use *secondary delay* and *propagated delay* synonymously, as for unbiased measuring of *delay-tolerance*, no recovery actions are considered. Delays are propagated until they are absorbed by idle times or the working day ends.

As delays belonging to deadheads are of least interest for the passengers and for transportation companies, propagated delays are only measured if they belong to timetabled trips/service trips. Therefore we use the following measure for *delay-tolerance*: We quantify the time (in seconds) each timetabled trip is starting behind schedule. If a timetabled trip is starting punctually, this time is zero. Then we calculate the expected length of a secondary delay per timetabled trip *E(SD)* as average over all these starting time deviations. Which is – in our case – the same as the average length of propagated delays per service trip. The goal of increasing delay-tolerance within a schedule is to minimize propagation of delays through the schedule.

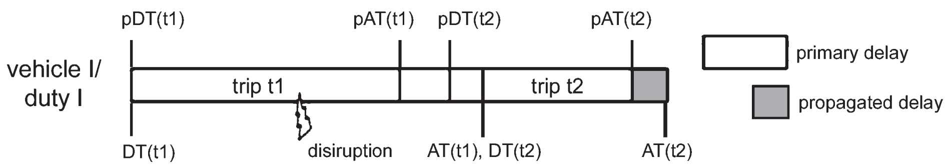

Figure 1 Primary and secondary/propagated delay

3. State-of-the-art approaches for vehicle and crew scheduling

For a recent survey on models for the VSP we refer to Bunte and Kliewer (2009). Recent comparative and computational tests of different heuristic approaches for vehicle scheduling can be found in Pepin et al. (2009).

Freling (1997) proposes the first integrated treatment of vehicle and crew scheduling. Based on this approach, Huisman (2004) and Huisman et al. (2005) propose the first general mathematical formulation for the integrated vehicle and crew scheduling problem with multiple depots. Gintner et al. (2006) and Steinzen et al. (2010) use the time-space network proposed by Kliewer et al. (2002) and Kliewer et al. (2006) for the MD-VSP and adapt it to the MD-IVCSP. In order to solve the model they use the general solution scheme proposed in Freling (1997). The solution approach described in Gintner et al. (2006) and in Steinzen et al. (2010), respectively, provides the basis for the work described in this paper.

In the context of public bus transport, there exist only a few approaches that consider robust vehicle scheduling and/or robust crew scheduling. In Huisman et al. (2004) a dynamic vehicle scheduling problem is presented, in which vehicle schedules are not previously constructed for a whole day, but are generated *online* during execution in order to cope with disruptions. However, because of the interdependencies with other planning phases in most practical cases such approaches are difficult to implement (cf. Huisman and Wagelmans (2006) for a discussion about dynamic vehicle and crew scheduling). In Huisman and Wagelmans (2006) the dynamic approaches are validated against a simple *offline* policy in which an equal fixed minimum buffer time must be adhered between all trips that are executed by the same vehicle. This policy is realized with the model for vehicle and crew scheduling proposed in Huisman et al. (2005). In Kramkoswki et al. (2009) an offline approach for delay-tolerant vehicle scheduling is presented. The authors use the metaheuristic simulated annealing for noisy environments to compute delay-tolerant vehicle schedules. In literature there exists no further investigation about improving delay-tolerance of both vehicle and crew schedules in public bus transport.

4. Basic solution framework

In this work we use models for vehicle scheduling based on an aggregated time-space network (TSN) from Kliewer et al. (2002) and Kliewer et al. (2006), respectively. The multi-depot vehicle scheduling problem is modeled as multi-commodity minimum-cost-flow problem in a TSN with one network layer for each depot-vehicle type combination. Each layer contains possible vehicle activities (trips, pull-out/in, deadheads, waiting) modeled as arcs between time-space nodes. The time-space nodes correspond to possible arrivals and departures at one station/depot. This modeling approach is adapted to the IVCSP by constructing time-space networks also for duty generation (cf. Gintner et al. (2006), Steinzen et al. (2010)).

Within the sequential approach and the partial-integrated approach, column generation is used to generate duties. The integrated model is solved by applying column generation in combination with Lagrangian relaxation. A subgradient method is used to solve the Lagrangian dual approximately (cf. Steinzen et al. (2010)). The column generation pricing problem is formulated as a resource constrained shortest path problem in which all work regulations concerning duty feasibility are considered. In contrast to Steinzen et al. (2010), we do not use a decomposed pricing strategy, but a task-based model with levels to generate new duties. The resource constraint shortest path problems are solved with the label setting algorithm described in Desrosiers et al. (1995).

5. Increasing delay-tolerance by redistributing buffer times based on the vehicle scheduling solution

Due to the time-space network structure an optimal network flow solution represents a bundle of feasible, cost-optimal vehicle schedules. Within this section we propose approaches to decompose this flow solution in order to increase delay-tolerance of the vehicle schedule. The main idea of the approaches is to redistribute the total amount of buffer times in the schedule without losing any cost-optimality. Using traditional sequential planning methods, the vehicle schedule serves as an input for the crew scheduling stage (see figure 2, according to Gintner et al. (2008)).

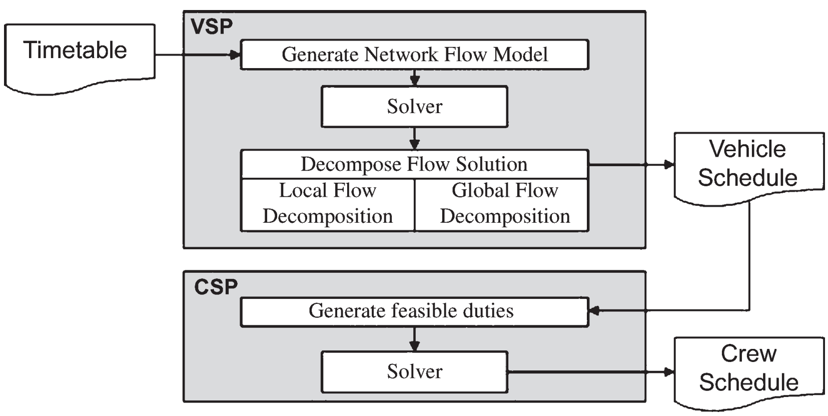

Figure 2 Sequential solution approach

5.1. Local Flow Decomposition – using no historical information about delays

As described in Kliewer et al. (2006) one simple local decomposition strategy for each node of the time-space network is First-In-First-Out (FIFO). It is obvious, that a decomposition based on FIFO tends to result in equal idle and buffer times for each vehicle. Thus, we use FIFO decomposition as a basic decomposition strategy. In contrast, a Last-In-First-Out (LIFO) decompostion strategy results in many short and many long buffer times.

5.2. Global Flow Decomposition – using historical information about delays

The simple FIFO decompostion scheme described above only makes local decisions. Hence, using this local strategy not all possible vehicle schedules can be considered in order to generate a delay-tolerant schedule. Therefore we propose a global flow decomposition scheme using information about the delay-tolerance of a bundle of cost-optimal vehicle schedules. Within these cost-optimal schedules we want to find the one that has got the highest delay-tolerance. As stated in section 2 our goal is to minimize the propagation of delays through the schedule. Hence, we propose the following (non) robustness-measure c_u in order to evaluate the delay-intolerance for each vehicle u that serves a number of T timetabled trips t:

$$c_u = \sum_{i=1}^{T-1} p(t_i, t_{i+1})$$

with $p(t_i, t_{i+1})$ defined as:

$$p(t_i, t_{i+1}) = \max\left\{0, \mathrm{pAT}(t_i) + \mathrm{PD}(t_i) + p(t_{i-1}, t_i) - \mathrm{pDT}(t_{i+1})\right\}$$

Where $p(t_0, t_1) = 0$, as this defines the propagated delay of the first trip, $pAT(t_i)$ is the planned arrival time of trip i, $PD(t_i)$ is the expected primary delay of trip i, and $pDT(t_{i+1})$ is the planned departure time of trip $i+1$. The expected primary delay of a trip has to be computed from historical delay data.

Notice, that $p(t_i, t_{i+1})$ corresponds to the secondary/propagated delay of trip $i+1$. Consequently, c_u corresponds to the sum of propagated delays within vehicle duty u. In order to find a vehicle schedule with minimum propagated delays within the cost-optimal network flow solution, we formulate the following set partitioning problem:

$$\sum_{d \in D} \sum_{u \in U^d} c_u^d x_u^d \rightarrow \min \qquad (1.1)$$

$$\text{s.t.} \sum_{u \in U^d(a)} x_u^d = f_a^d \qquad \forall d \in D, \forall a \in A^d \quad (1.2)$$

$$x_u^d \in \{0,1\} \quad \forall d \in D, \forall u \in U^d \quad (1.3)$$

Where D corresponds to the set of depot-vehicle type combinations. U_d denotes the set of all feasible vehicle duties that can be operated from depot d. Furhtermore, $U_d(a)$ denotes the set of all vehicle duties which contain the trip/task represented by arc $a \in A^d$ within the time-space-network that has been set up for depot d. f^d_a corresponds to the flow value on arc $a \in A^d$. We introduce binary decision variables x^d_u that indicate whether vehicle duty $u \in U^d$ is selected or not.

The objective (1.1) minimizes the sum of propagated delays within the vehicle schedule, hence increasing the delay-tolerance of the schedule. Constrains (1.2) ensure that each network arc is covered exactly by the required number of vehicles.

As the number of possible, feasible vehicle duties is huge we solve the problem with column generation. The initial solution is obtained by a local flow decomposition scheme such as FIFO. New columns/vehicle duties are generated by solving resource constraint shortest path problems with the label setting algorithm described in Desrosiers et al. (1995).

6. Increasing delay-tolerance by redistributing buffer times based on the crew scheduling solution

The approaches described in section 5 have the drawback that crew duties are based on a fixed underlying vehicle schedule. Hence, redistributing buffer times within the vehicle schedule without increasing planned costs for vehicles maybe induces a small increase of costs for the crew schedule. In order to avoid this drawback, we adapt the partial-integrated scheduling approach proposed by Gintner et al. (2008). As aforementioned, due to the underlying network model, we obtain a bundle of optimal vehicle schedules, implicitly given by the solution flow. Instead of decomposing the solution flow during vehicle scheduling, we give this degree of freedom to the crew scheduling phase. Figure 3 shows the interaction between vehicle and crew scheduling (according to Gintner et al. 2008). Since each decomposition of the optimal flow of the VSP-network corresponds to an optimal vehicle schedule and the crew scheduling problem is solved based on this bundle of optimal solutions, an optimal vehicle schedule can always be built afterwards. For a detailed description of the partial-integrated scheduling approach we refer to Gintner et al. (2008). In this section we will focus on the description of our approaches to increase delay-tolerance of both vehicle and crew schedules using different decomposition schemes in order to redistribute buffer times based on the crew scheduling solution.

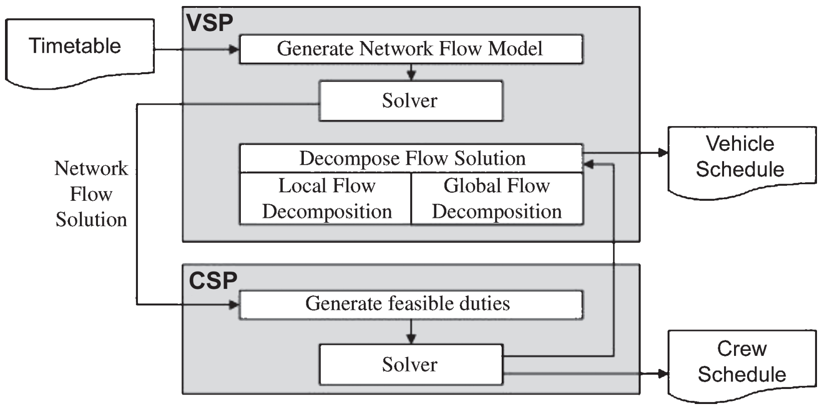

Figure 3 Partial-integrated solution approach

6.1. Local Flow Decomposition – using no historical information about delays

Based on the crew scheduling solution we derive all vehicle blocks that are necessary to operate the crew schedule. Recall that vehicle blocks are sequences of served trips between two layovers at a depot that start with a pull-out and end with a pull-in trip. These vehicle blocks have to be merged to build vehicle duties taking delay-tolerance into consideration. We use a heuristic procedure to combine vehicle blocks to delay-tolerant vehicle duties: For each depot-vehicle type combination we sort all vehicle blocks according to increasing departure time of the pull-out trips. Starting with the vehicle block with the lowest departure time, vehicle blocks are connected, if a required minimum buffer time fits between arrival time of one vehicle block (vehicle block combination) and departure time of another vehicle block (vehicle block combination). This step is repeated until no additional connection between vehicle blocks (vehicle block combinations) is possible. However, forcing a specific minimum buffer time might increase the number of needed vehicles used to operate the schedule (- there will be no increase in the operational costs of the vehicle schedule). As we are interested to maintain a cost-optimal vehicle schedule, we repeat the procedure described above, starting with a required minimum buffer time of 0 minutes and increase the buffer time by 1 minute as long as the vehicle number is minimal. Notice, that due to the network flow solution we know the minimal fleet size for each depot-vehicle type combination (as this can be seen on the flow value of the corresponding flow circulation arc).

6.2. Global Flow Decomposition – using historical information about delays

As stated in the previous section, within partial-integrated scheduling vehicle blocks are obtained from the crew scheduling solution and have to be connected to vehicle duties. Using historical information about (primary) delays we propose a decomposition scheme in order to find a cost-optimal vehicle schedule that has got a high delay-tolerance and minimizes the propagation of delays, respectively. In contrast to the vehicle schedule generated with the approach proposed in section 5.2, the computed vehicle schedule will not influence the cost of the crew scheduling solution. Using the robustness-measure described in section 5.2, firstly, we compute the delay-intolerance c_b for each vehicle block b. In order to connect vehicle blocks considering their delay-intolerance, we set up the following network.

Vehicle blocks are represented as arcs, connecting the corresponding pull-out node and pull-in node. Each pull-out node represents a departure event from a depot, and each pull-in node represents an arrival event at a depot. We insert arcs representing possible connections with other vehicle blocks between each pull-in node and all pull-out whose departure time is later than the arrival time of the pull-in node. These connecting arcs are provided with cost computed as follows: Let b_i and b_j be vehicle blocks with pull-out time of b_j is later than pull-in time of b_i, then

$$p(b_i, b_j) = \max\left\{0, \mathrm{pAT}(b_i) + c_b - \mathrm{pDT}(b_j)\right\}$$

describes the possible propagated delay between vehicle block b_i and vehicle block b_j. Finally, we add a source node which is connected to all pull-out nodes, and a sink node which is connected with all pull-in nodes. A flow circulation arc connects the sink node with the source node.

Based on the described network structure we set up a minimum cost-flow problem for each depot-vehicle type combinations in order to minimize the propagation of delays through the schedule. The resulting network flow model contains one network layer for each depot vehicle-type combination where an integer flow variable is defined for the circulation arc and 0/1 variables for all other arcs. The flow value on the arcs representing vehicle blocks is fixed to one. The flow value on the circulation arc is fixed to the minimum number of vehicles in the cost-optimal solution. The solution vector describes the flow solution in each network layer with minimal propagated delay and in addition represents a cost-optimal solution. Each flow unit starting in the source node, flowing through the network arcs, and returning back through the circulation arc into the source node represents a vehicle.

In order to increase the solution space we realize a similar approach by constructing a network based on pieces of work instead of vehicle blocks. As a vehicle block usually contains several pieces of work, this approach allows us to consider more possibilities to build a delay-tolerant schedule. Recall that pieces of work can start and end outside a depot, namely at each station where a driver may change his vehicle. Thus, nodes within the network represent start time and end time of the pieces of work. Pieces of work are represented as arcs, connecting the corresponding start and end nodes.

7. Increasing delay-tolerance by increasing required minimum buffer time for connecting trips

All aforementioned approaches have in common that the delay-tolerance of vehicle schedule and crew schedule is improved by redistributing the total amount of buffer time. As a consequence the planned costs for the vehicle schedule or even for both vehicle schedule and crew schedule remain the same compared to pure cost-efficient scheduling. However, the delay-tolerance can also be improved by introducing additional buffer times. Again, we consider two types of approaches: using no information about delays, and approaches using historical delay data.

One simple method that uses no historical delay information is to introduce a *global buffer time*: After each service trip the same minimum buffer time is required before another trip can be operated. With increasing buffer time this method obviously leads to extremely robust schedules to the disadvantage of planned costs. Further, duties can be created that require an additional buffer time after each n^{th} (service) task. Due to our solution framework it is possible to define such rules for state propagation within the label setting pricing algorithm.

If historical delay data is available, such as primary delay scenarios, buffer time between service trips can be enforced because of different strategies. Firstly, buffer time can be added after each service trip in order to cope with the *worst-cas*e primary delay out of a set of delay scenarios. However, as this is a costly strategy, only *x% of the worst case* primary delays could be considered. Further, after each service trip can be required buffer time to absorb the *average* primary delay of this service trip within a set of delay scenarios.

8. Computational results

We tested our approaches on real-world instances from public transit companies in Germany. All tests were performed on a Dell OptiPlex 755 personal computer running Windows XP with Intel Core 2 Duo 3.0 GHz and 4 GB of main memory. Our solution approaches are implemented in C# and have been compiled using the .NET framework version 2.0.5027. We use ILOG CPLEX 12.2 for computing integer solutions. According to Huisman (2004) we consider five different types of duties when solving crew scheduling problems: one tripper type with one piece of work between 30 minutes and 5 hours, and four types consisting of two pieces of work. We use the same cost settings as described in Steinzen et al. (2010): The main objective is to minimize the total number of vehicles and drivers. Operational cost minimization is a secondary objective. Therefore fixed cost for each vehicle and duty are set to 1,000 while variable cost are set to smaller values; variable vehicle cost are set to 1 for each minute a vehicle is outside the depot whereas cost for a minute crew working time are set to 0.1. Recall, that when using methods proposed in section 5 and 6, the objective is to minimize the propagated delay. Thus we compare different approaches with respect to *E(SD)* and *planned cost*. E(SD) for each vehicle and crew schedule is computed as average over 500 independent simulation runs. The simulation uses a probability distribution for primary delays derived from real-world delay scenarios (see Huisman et al. (2004)).

Table 1 Comparison of solution approaches with respect to planned cost and delay-tolerance E(SD) – instance A

Solution Approach	*#v*	*VS costs*	*#d*	*CS costs*	*VS+CS costs*	*costs in %*	*E(SD) in sec*	*E(SD) in %*
Sequential	17	26514	41	42076.6	68590.6	x	84.1	x
Global Flow Decomposition	17	26514	41	42258.4	68772.4	100.3	74.1	88.1
Buffer average	17	26516	42	43075	69591	101.5	74.6	88.7
Buffer 1	17	26589	41	42077.3	68666.3	100.1	70.6	83.9
Buffer 2	17	26717	45	46095.4	72812.4	106.2	55.1	65.4
Buffer 3	18	28093	54	55153.6	83246.6	121.4	49.8	59.3
Buffer 4	19	29315	55	56175.3	85490.3	124.6	50.0	59.4
Partial-integrated	17	26514	39	39947	66461	96.9	79.7	94.8
Local Flow Decomposition	17	26514	39	39947	66461	96.9	74.6	88.7
Global Flow Decomposition	17	26514	39	39947	66461	96.9	71.6	85.1
Global Flow Decomposition Piece	17	26514	39	39947	66461	96.9	71.4	84.9
Buffer average	17	26516	40	40974.6	67490.6	98.4	63.6	75.7
Buffer 1	17	26589	41	42019.8	68608.8	100.0	61.1	72.6
Buffer 1 + Global Flow Dec	17	26589	41	42019.8	68608.8	100.0	57.7	68.6
Buffer 1 + Global Flow Dec Piece	17	26589	41	42019.8	68608.8	100.0	57.7	68.6
Buffer 2	17	26717	45	46037	72754	106.1	56.9	67.6
Buffer 2 + Global Flow Dec	17	26717	45	46037	72754	106.1	54.9	65.3
Buffer 2 + Global Flow Dec Piece	17	26717	45	46037	72754	106.1	54.9	65.3
Buffer 3	18	28093	53	54098	82191	119.8	41.5	49.3
Buffer 3 + Global Flow Dec	18	28093	53	54098	82191	119.8	38.7	46.0
Buffer 3 + Global Flow Dec Piece	18	28093	53	54098	82191	119.8	38.5	45,8
Buffer 4	19	29315	55	56147.6	85462.6	124.6	44.9	53.3
Integrated	17	27659	29	30141.6	57800.6	84.3	90.5	107.6
Buffer average	17	27459	30	31121.3	58580.3	85.4	63.9	75.9
Buffer 1	17	27615	29	30119.4	57734.4	84.2	65.1	77.4
Buffer 2	17	27809	29	30172.2	57981.2	84.5	48.0	57.1
Buffer 3	18	29318	30	31202.5	60520.5	88.2	46.8	55.7
Buffer 4	19	30670	34	35245	65915	96.1	51.0	60.6

Table 2 Comparison of solution approaches with respect to planned cost and delay-tolerance E(SD) – instance B

Solution Approach	*#v*	*VS costs*	*#d*	*CS costs*	*VS+CS costs*	*costs in %*	*E(SD) in sec*	*E(SD) in %*
Sequential	54	78972	88	91402.3	170374.3	x	80.3	x
Global Flow Decomposition	54	78972	88	91495.8	170467.8	100.1	78.9	98.3
Buffer average	54	80324	92	94823.9	175147.9	102.8	76.1	94.8
Buffer 1	55	81806	93	95850.1	177656.1	104.3	64.6	80.5
Buffer 2	56	84321	99	102021.8	186342.8	109.4	50.4	62.8
Buffer 3	58	87035	101	104127.3	191162.3	112.2	48.2	60.0
Buffer 4	58	87294	105	113303.9	200597.9	117.7	46.6	58.0
Partial-integrated	54	78972	85	87691.7	166663.7	97.8	79.1	98.5
Local Flow Decomposition	54	78972	85	87691.7	166663.7	97.8	70.6	87.9
Global Flow Decomposition	54	78972	85	87691.7	166663.7	97.8	68.5	85.3
Global Flow Decomposition Piece	54	78972	85	87691.7	166663.7	97.8	68.5	85.3
Buffer average	54	80324	88	91812.4	172136.4	101.0	70.4	87.7
Buffer 1	55	81806	92	94871.1	176677.1	103.7	68.1	84.8
Buffer 1 + Global Flow Dec	55	81806	92	94871.1	176677.1	103.7	67.3	83.8
Buffer 1 + Global Flow Dec Piece	55	81806	92	94871.1	176677.1	103.7	66.9	83.3
Buffer 2	56	84321	96	99049.1	183370.1	107.6	54.1	67.4
Buffer 2 + Global Flow Dec	56	84321	96	99049.1	183370.1	107.6	51.2	63.8
Buffer 2 + Global Flow Dec Piece	56	84321	96	99049.1	183370.1	107.6	51.1	63.6
Buffer 3	58	87035	98	100780.0	187815.0	110.2	49.4	61.5
Buffer 3 + Global Flow Dec	58	87035	98	100780.0	187815.0	110.2	47.3	58.9
Buffer 3 + Global Flow Dec Piece	58	87035	98	100780.0	187815.0	110.2	46.7	58.2
Buffer 4	58	87294	101	104149.4	191443.4	112.4	47.5	59.1
Integrated	54	81389	80	82916.2	164223.2	96.4	84.0	104.6
Buffer average	54	82586	82	85044.6	167630.6	98.4	65.2	81.2
Buffer 1	55	85186	87	89816.1	175002.1	102.7	58,2	72.5
Buffer 2	56	87092	91	94313.7	181405.7	106.5	55.3	68.9
Buffer 3	58	89741	91	94402.1	184143.1	108.1	50.9	63.4
Buffer 4	58	89962	92	95396.2	185358.2	108.8	48.6	60.5

Table 1 and table 2 give exemplary computational results for a small instance with 211 trips, 1 depot and 1 vehicle type (instance A) and for a medium size instance with 580 trips (instance B). The structure of both tables is the same: The first column reports the used solution approach. The tables are divided into three parts according to the used solution approaches (*sequential, partial-integrated, and integrated*). The first row of each part reports results for scheduling without considering delay-tolerance of the computed schedules. E.g. *sequential* describes the traditional sequential planning approach using a local FIFO decomposition scheme in order to obtain a vehicle schedule before scheduling crews. *Global flow decomposition* labels approaches using historical information about delays. *Global flow decomposition piece* labels the approach based on pieces of work that is described in section 6.2. *Buffer* denotes results computed with approaches requiring a fixed buffer time for connecting trips. Consequently, *buffer + global flow decomposition (piece)* denotes a combination of both methods.

In general, the results show that with little increase in planned-cost, schedules can be generated that are more delay-tolerant. Delay-tolerance of schedules can even be improved without losing any cost-optimality. However, there is a trade-off between cost-optimal solutions and delay-tolerant one. In addition, the results show that there is an efficiency gain if vehicle and crew scheduling are (partial-)integrated.

However, future work has to be done in the area of bi-objective models and/or methods especially within integrated planning, as this planning method offers the greatest degrees of freedom for scheduling.

References

Bunte, S., Kliewer, N. (2009) An overview on vehicle scheduling models. *Public Transport* 1(4), 299-317, Springer Berlin/Heidelberg, Germany

Desrosiers, J., Dumas, Y., Solomon, M. M., Soumis, F. (1995) Time constrained routing and scheduling. *Handbooks in Operations Research and Management Science* 8, chapter 2, 35-139, Elsevier Science, Amsterdam

Freling, R. (1997) Models and Techniques for Integrating Vehicle and Crew Scheduling, *Ph.D Thesis*, Erasmus University of Rotterdam, The Netherlands

Gintner, V., Steinzen, I., Suhl, L. (2006) A time-space network based approach for integrated vehicle and crew scheduling in public transport. *Proceedings of the 11th meeting of the EURO working group on Transportation (EWGT)*. Polotechnic of Bari, Italy

Gintner, V., Kliewer, N., Suhl, L. (2008) A crew scheduling approach for public transit enhanced with aspects from vehicle scheduling. *Computer-Aided Systems in Public Transport. Lecture Notes in Economics and Mathematical Systems*, Vol. 600, Springer

Huisman, D. (2004) Integrated and dynamic vehicle and crew scheduling. *Ph.D. Thesis*, Erasmus University of Rotterdam, The Netherlands.

Huisman, D., Freling, R., Wagelmans, A.P. (2004) A robust solution approach to the dynamic vehicle scheduling problem. *Transportation Science* 38, 447-458

Huisman, D., Freling, R., Wagelmans, A.P. (2005) Multiple-depot integrated vehicle and crew scheduling. *Transportation Science* 39 (4), 491-502.

Huisman, D., Wagelmans, A.P. (2006) A Solution Approach for Dynamic Vehicle and Crew Scheduling, *European Journal of Operational Research* 172, 453-471.

Kliewer, N., Mellouli, T., Suhl L. (2002) A new solution model for multi-depot multi-vehicle type vehicle scheduling in (sub)urban public transport. *Proceedings of the 13th Mini-EURO Conference and the 9th meeting of the EURO working group on Transportation (EWGT)*. Polotechnic of Bari, Italy

Kliewer, N., Mellouli, T., Suhl, L. (2006) A time-space network based exact optimization model for multiple-depot bus scheduling. *European Journal of Operations Research* 175(3), 1616-1627

Kramkowski, S., Kliewer, N., Meier, Ch. (2009) Heuristic Methods for Increasing Delay-Tolerance of Vehicle Schedules in Public Bus Transport. *MIC 2009 Proceedings, Metaheuristic International Conference VIII*, Hamburg, Germany

Pepin, A.-S., Desaulniers, G., Hertz, A., Huisman, D. (2009) A comparison of five heuristics for the multiple depot vehicle scheduling problem. *Journal of scheduling* 12 (1), 17-30

Steinzen, I., Gintner, V., Suhl, L., Kliewer, N. (2010) A Time-Space Network Approach for the Integrated Vehicle- and Crew-Scheduling Problem with Multiple Depots. *Transportation Science* 44(3), 367-382