Applying an Integrated Approach to Vehicle and Crew Scheduling in Practice

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

ERIM Report Series *Research in Management*	
ERIM Report Series reference number	ERS-2000-31-LIS
Publication status / version	draft / version June 2000
Number of pages	20
Email address corresponding author	Huisman@few.eur.nl
Address	Erasmus Research Institute of Management (ERIM) Rotterdam School of Management / Faculteit Bedrijfskunde Erasmus Universiteit Rotterdam PoBox 1738 3000 DR Rotterdam, The Netherlands Phone: # 31-(0) 10-408 1182 Fax: # 31-(0) 10-408 9020 Email: info@erim.eur.nl Internet: www.erim.eur.nl

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

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

REPORT SERIES
RESEARCH IN MANAGEMENT

BIBLIOGRAPHIC DATA AND CLASSIFICATIONS		
Abstract	This paper deals with a practical application of an integrated approach to vehicle and crew scheduling, that we have developed previously. Computational results have shown that our approach can be applied to problems of practical size. However, application of the approach to the actual problems that one encounters in practice, is not always straightforward. This is mainly due to the existence of particular constraints that can be regarded as \house rules" of the public transport company under consideration. In this paper we apply our approach to problems of individual bus lines of the RET, the Rotterdam public transport company, where particular constraints should be satisfied. Furthermore, we investigate the impact of allowing drivers to change vehicle during a break. Currently, the rule at the RET is that such changeovers are only allowed in split duties; they are never allowed in other type of duties. We show that it is already possible to save crews if for the non-split duties, restricted changeovers are allowed.	
Library of Congress Classification (LCC)	5001-6182	Business
	5201-5982	Business Science
	HD 69.T54	Scheduling
Journal of Economic Literature (JEL)	M	Business Administration and Business Economics
	M 11 R 4	Production Management Transportation Systems
	C 6	Mathematical Methods and Programming
European Business Schools Library Group (EBSLG)	85 A	Business General
	260 K 240 B	Logistics Information Systems Management
	250 A	Mathematics
Gemeenschappelijke Onderwerpsontsluiting (GOO)		
Classification GOO	85.00	Bedrijfskunde, Organisatiekunde: algemeen
	85.34 85.20	Logistiek management Bestuurlijke informatie, informatieverzorging
	31.80	Toepassingen van de wiskunde
Keywords GOO	Bedrijfskunde / Bedrijfseconomie	
	Bedrijfsprocessen, logistiek, management informatiesystemen	
	Algoritmen, Scheduling, Vervoersmodellen, Roosters	
Free keywords	Vehicle scheduling, crew scheduling, integrated planning, column generation, Lagrangian relaxation	
Other information		

Applying an Integrated Approach to Vehicle and Crew Scheduling in Practice

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

Econometric Institute, Erasmus University Rotterdam, PO BOX 1738,
NL-3000 DR Rotterdam, The Netherlands

Abstract. This paper deals with a practical application of an integrated approach to vehicle and crew scheduling, that we have developed previously. Computational results have shown that our approach can be applied to problems of practical size. However, application of the approach to the actual problems that one encounters in practice, is not always straightforward. This is mainly due to the existence of particular constraints that can be regarded as "house rules" of the public transport company under consideration.

In this paper we apply our approach to problems of individual bus lines of the RET, the Rotterdam public transport company, where particular constraints should be satisfied. Furthermore, we investigate the impact of allowing drivers to change vehicle during a break. Currently, the rule at the RET is that such changeovers are only allowed in split duties; they are never allowed in other type of duties. We show that it is already possible to save crews if for the non-split duties, resticted changeovers are allowed.

1 Introduction

Commercially successful computer packages use a sequential approach for vehicle and crew scheduling. Sometimes integration is dealt with at the user level (e.g., see Darby-Dowman et al. (1988)) or crew considerations are taken into account in the vehicle scheduling phase (for example in HASTUS, see Rousseau and Blais (1985)). In the operations research literature, only a few publications address a simultaneous approach to vehicle and crew scheduling. For an overview, we refer to Freling et al. (2000). None of the publications mentioned there makes a comparison between simultaneous and sequential scheduling. Furthermore, to the best of our knowledge, no integrated model has been applied to a practical situation before.

In Freling et al. (2000) some models and algorithms for integration of vehicle and crew scheduling were discussed and applied to problems of practical size. The test problems were based on real-life problems from the RET, the public transport company in the city of Rotterdam, the Netherlands. In these test problems, we omitted certain complicating constraints that are specific to the RET. In this paper, we will apply our approach to the actual RET problems. Although the RET also provides service on tram and metro lines, only bus lines will be considered. For bus lines we can expect a larger benefit of the integration, because the relative difference between crew and vehicle costs is higher than in tram and metro scheduling. Moreover, tram and metro scheduling problems are more restricted because of constraints with respect to the capacity of the rail track and the capacity at the endpoints of the lines.

We will compare our integrated approach with a sequential approach. Furthermore, we will investigate the impact of allowing drivers to change vehicle during a break. Currently, the rule at the RET is that such *changeovers* are only allowed in split duties; they are never allowed in other type of duties. We show that it is already possible to save crews if for the non-split duties, resticted changeovers are allowed.

The paper is organized as follows. Sect. 2 describes the situation which holds for almost all bus lines at the RET and especially for the lines we consider in this paper. In Sect. 3, we discuss a sequential approach that takes all relevant RET constraints into account. The integrated approach that we use, is discussed in Sect. 4. The paper is concluded with a computational study (Sect. 5). The main objective of this study is threefold:

1. We show that our integrated approach can be applied to a practical situation.
2. We make a comparison between the results of the sequential and the integrated approach.
3. We look at the effect of allowing (restricted) changeovers.

2 Problem Description at the RET

In Fig. 1 we show the relation between four operational planning problems at the RET.

Decisions about which routes or *lines* to operate and how frequently, are determined by local and regional authorities and are given to the RET. Also known are the travel times between various points on the route. Based on the lines and frequencies, timetables are determined resulting in *trips* with corresponding start and end locations and times. The second planning process is vehicle scheduling, which consists of assigning vehicles to trips, resulting in a schedule for each vehicle or *vehicle blocks.* On each vehicle block a sequence of tasks can be defined, where each task needs to be assigned to a working period for one crew (a *crew duty* or *duty)* in the crew scheduling process. The feasibility of a duty is dependent on a set of collective agreements and labor rules, that refer to sufficient rest time, et cetera. Crew scheduling is short term crew planning (one day) for assigning crews to vehicles, while the crew rostering process is long term crew planning (e.g. half a year) for constructing rosters from the crew duties.

In Subsect. 2.1, we discuss the definitions and restrictions at the RET and in Subsect. 2.2, we focus on one particular restriction, which will play a major role in the rest of this paper.

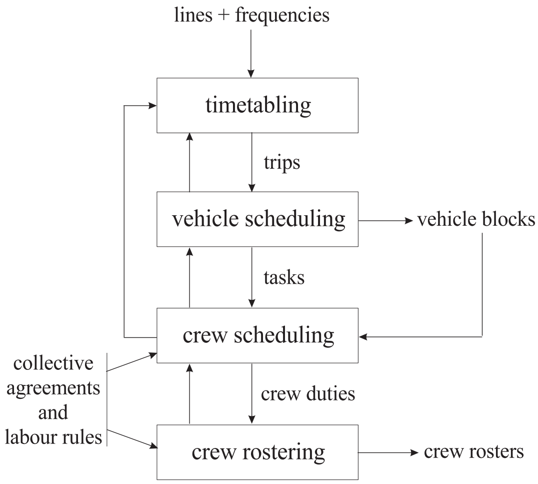

Fig. 1. Four related planning problems

2.1 Definitions and Restrictions

At the RET the whole planning process is solved line-by-line, so there is no inter-lining and the size of the problems is relatively small (up to 259 trips). Almost all lines have two locations (lets call them A and B) and all the trips are from A to B or from B to A. In this paper we only consider lines with this property.

For every line there is one given depot and all buses must start and end in that depot. A *deadhead* is the driving of a vehicle without passengers. There are three kinds of deadheads: from the depot to the start of a trip, from the end of a trip to the depot and between two trips (from one endpoint of the line to the other one). The workday of one driver is called a *(crew) duty*. There are many different types of duties, for example early duties, late duties and split duties. Every duty consists of two pieces with in between a break. A *piece* consists of consecutive *tasks* (trips, layovers and deadheads) performed by one driver on the same bus. A *layover* is the time between two consecutive trips that a driver with bus waits at a start or end location of the line. For the layover there are several restrictions:

- after every trip and deadhead there is a layover of at least two minutes;
- there is a minimal layover in a *round-trip*, where a round-trip consists of two consecutive trips from location A to B and back to A; this is the so-called *round-trip-condition*, that will be dicussed in more detail in Subsect. 2.2;
- the total layover time in a duty is at least 10% of the length of the duty;
- if the longest piece in a split duty is more than 4 hours, then there is at least once a layover of at least 10 minutes.

All duties start or end at the depot or at so-called relief locations. A *relief location* is the start or the end location of the line. In some cases both the start and the end location is a relief location and in other cases only one of them is a relief location. Duties starting at the depot always start with a *sign-on* time and the other duties start with a *relief* time. Duties ending at the depot end with a *sign-off* time and the other duties end after a layover. These times are fixed and known in advance. The breaks are only allowed at relief locations. A break is only necessary in duties that have a length greater than a certain minimum.

Furthermore there are restrictions for all 8 types of duties with respect to the earliest and latest starting time, the latest ending time, the minimum length of the break, the maximum spread and the maximum working time. On Sundays there are no split duties. For more details we refer to Huisman (1999).

Currently changeovers are only allowed in split duties and under no circumstances in other duties. In this paper we compare this situation with the variant where a changeover is also allowed during a break of a non-split duty, provided that there is enough time to change vehicle. In that case the first piece ends with a layover and the second piece starts with the relief time. Of course, the length of the last layover of the first piece is then at least the relief time, because the driver who takes over the first bus needs this amount of time. As a consequence there is continuous attendance of the vehicle, i.e., there is always a driver present when the bus is outside the depot.

2.2 Round-trip-condition

In this subsection, we discuss the most important and complicated constraint of the RET: the round-trip-condition. As mentioned before, the round-trip-condition is a restriction on the total layover in a round-trip (two consecutive trips such that a driver drives from location A via location B back to location A). The round-trip-condition requires that the total layover at the locations A and B in one round-trip is at least 10 minutes if the round-trip is 60 minutes or longer and 15% of the travel time if the round-trip is shorter than 60 minutes with a minimum of 5 minutes. It is allowed to violate this condition at most once in every crew duty, but only if the total layover is at least 5 minutes.

In advance, we can construct combinations of three trips that violate the round-trip-condition. We show this in Fig. 2. We use these combinations in the sequential approach as well as in the integrated approach.

In a sequential approach it is not possible to incorporate the possibility of violating the round-trip-condition at most once in every crew duty. The reason is that we have to deal with the round-trip-condition in the Vehicle Scheduling Problem (VSP) and at that stage we do not know anything yet about crew duties. Hence, if we violate the round-trip-condition more than once in the VSP, we may later on end up with an infeasible solution to the

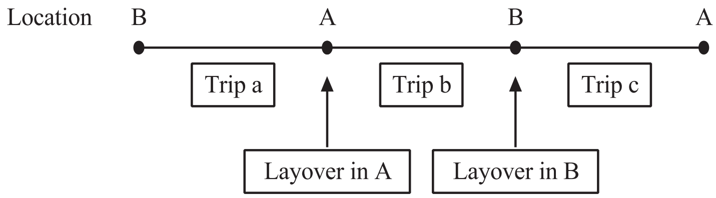

Fig. 2. Round-trip-condition

Crew Scheduling Problem (CSP), because some crew duty has more than one violation of the round-trip-condition.

As we will see in Sect. 4, in our integrated approach, we can exploit the possibility that the round-trip-condition may be violated once in every crew duty. This is actually the main reason why we can save vehicles by using an integrated approach instead of a sequential one.

3 Sequential Approach

In this section, we give a sequential approach for solving the Vehicle Scheduling Problem (VSP) and the Crew Scheduling Problem (CSP). First we discuss in Subsect. 3.1 the VSP with round-trip-condition and after that we discuss in Subsect. 3.2 the CSP. For both problems we give a mathematical model and a solution method.

3.1 Vehicle Scheduling

Let $N = \{1, 2, ..., n\}$ be the set of trips, numbered according to increasing starting time, and let $E = \{(i, j) \mid i < j,\, i, j \text{ compatible}, i \in N, j \in N\}$ be the set of arcs corresponding to deadheads and layovers. The nodes s and t both represent the depot. We define the vehicle scheduling network $G = (V, A)$, which is an acyclic directed network with nodes $V = N \cup \{s, t\}$, and arcs $A = E \cup (s \times N) \cup (N \times t)$. A path from s to t in the network represents a feasible schedule for one vehicle, and a complete feasible vehicle schedule is a set of disjoint paths from s to t such that each node in N is covered. Let R be the set of combinations of trips that violate the round-trip-condition. Every combination consists of three trips, that we will refer to as a, b and c like in Fig. 2.

Let c_{ij} be the vehicle cost of arc $(i, j) \in A$, which is usually some function of travel and idle time. Furthermore, a fixed cost K for using a vehicle can be added to the cost of arcs (s, i) or (j, t) for all $i, j \in N$. For the remainder of this paper, we assume that the primary objective is to minimize the number of vehicles. This means that K is high enough to guarantee that this minimum number will be achieved.

Using decision variables y_{ij}, with $y_{ij} = 1$ if a vehicle covers trip j immediately after trip i, $y_{ij} = 0$ otherwise, the VSP with round-trip-condition can be formulated as follows:

$$\min \sum_{(i,j) \in A} c_{ij} y_{ij} \tag{1}$$

$$\sum_{j:(i,j) \in A} y_{ij} = 1 \qquad \forall i \in N, \tag{2}$$

$$\sum_{i:(i,j) \in A} y_{ij} = 1 \qquad \forall j \in N, \tag{3}$$

$$y_{ab} + y_{bc} \leq 1 \qquad \forall (a,b,c) \in R, \tag{4}$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A. \tag{5}$$

Constraints (2) and (3) assure that each trip is assigned to exactly one predecessor and one successor, that is, these constraints guarantee that the network is partitioned into a set of disjoint paths from s to t. Constraint (4) assures that the round-trip-condition is met, because the trips a, b and c cannot be assigned to the same vehicle.

When a vehicle has an idle time between two consecutive trips which is long enough to let it return to the depot, it does so. In that case the arc between the trips is called a *long arc;* the other arcs are called *short arcs.* In general, a bottleneck for solving VSP when using a network such as G may be the large size of those networks due to a large number of arcs in E. For the problems that we consider in this paper, however, the number of arcs does not cause serious complications.

We solve this model with subgradient optimization and Lagrangian relaxation, where we relax constraint (4). Then the remaining subproblem is a quasi-assignment problem, which is solved in every iteration of the subgradient optimization with an auction algorithm (see Freling, Paixão and Wagelmans Freling et al. (1995)). In this way we get a lower bound on the optimal solution. Furthermore, we also compute a feasible solution in every iteration by changing the solution of the subproblem such that the round-trip-condition is not violated anymore. This requires at least one more vehicle. We terminate the subgradient optimization if the gap between the lower bound and the value of the best feasible solution indicates that we have found an (almost) optimal solution.

3.2 Crew Scheduling

We solve the Crew Scheduling Problem (CSP) in two steps, first we generate all feasible duties and then we select the optimal duties by solving a set covering model.

Generation of Duties

After the computation of the optimal vehicle schedule we generate duties in three steps.

1. Definition of all relief points. A relief point is the point on the vehicle block, where a driver can have his/her break or can be relieved by another driver. Of course, the depot is also a relief point. For the other relief points the following two properties hold: first the location of the relief point is a relief location, and second the relief point is after a layover whose length is greater than or equal to the relief time.
2. Generation of all feasible pieces. A piece is the work between two relief points on the same vehicle block. Here we take into account a maximum length of the piece. We use these pieces also as *trippers*, i.e., one-piece duties.
3. Generation of all feasible duties. Two pieces with a break in between are combined to construct a duty. Here we take care of restrictions on the length of the duty, the length and position of the break, the maximum spread and so on. In some variants the two pieces must belong to the same vehicle, but in some other variants this restriction is not present.

Mathematical Formulation CSP

Let d_k be the cost of duty $k \in K$, where K is the set of all feasible duties, and define $K(i) \in K$ as the set of duties covering task $i \in I$. A task can be a trip or a deadhead. Consider binary decision variable x_k indicating whether duty k is selected in the solution or not. In the set covering formulation of the CSP, the objective is to select a minimum cost set of feasible duties such that each task is included in one of these duties. This is the following 0-1 linear program:

$$\min \sum_{k \in K} d_k x_k \tag{6}$$

$$\sum_{k \in K(i)} x_k \geq 1 \qquad \forall i \in I, \tag{7}$$

$$x_k \in \{0, 1\} \qquad \forall k \in K, \tag{8}$$

where constraints (7) assure that each task will be covered by at least one duty. The solution of the set covering problem can be translated into a solution of the CSP by simply deleting double trips. The most important advantage of using a set covering formulation instead of a set partitioning formulation is the fast computation time.

Step 0: Initialization
Generate a set of pieces such that each task can be covered by at least one piece. The initial set of columns consists of these pieces.
Step 1: Computation of lower bound
Solve a Lagrangian dual problem with the current set of columns.
Step 2: Generation of columns
Generate columns with negative reduced cost. If no such columns exist (or another termination criterion is satisfied), go to Step 3; otherwise, return to Step 1.
Step 3: Construction of feasible solution
Use all the columns generated in Step 0 and Step 2 to construct a feasible solution.

Fig. 3. Solution method for CSP

Solution Method CSP

The algorithm for solving the CSP is shown in Fig. 3.

For further details we refer the interested reader to Freling et al. (2000) and to Freling (1997).

4 Integrated Approach

In this section we discuss the integrated approach which we applied to the problems of the RET. First we describe in Subsect. 4.1 and 4.2 a model for the integrated Vehicle and Crew Scheduling Problem (VCSP) and a general solution method, respectively. The big difference between an integrated and a sequential approach is that it is not possible to generate all feasible duties in advance, because the number of feasible duties is too large. The problem is to generate the pieces, such that we can combine two pieces to a duty and in our case we have also to implement the round-trip-condition in the generation of pieces. We explain this in Subsect. 4.3 for the case where violation of the round-trip-condition is not allowed and in Subsect. 4.4 where violation is allowed at most once. Finally, we discuss in Subsect. 4.5 how we implement the restrictions with respect to relieving.

4.1 Mathematical Model VCSP

The mathematical formulation we propose for the VCSP is a combination of the quasi-assignment formulation for vehicle scheduling based on network $G = (V, A)$ defined in Subsect. 3.1, and a set partitioning formulation for crew scheduling. The quasi-assignment part assures the feasibility of vehicle schedules, while the set partitioning part assures that each trip is assigned to a duty and each deadhead task is assigned to a duty if its corresponding deadhead is part of the vehicle schedule. Before providing the mathematical formulation, we need to recall and introduce some notation. As before, N denotes the set of trips, K denotes the set of duties, and $A^s \subset A$ and $A^l \subset A$ denote the sets of short and long arcs, respectively. Furthermore, $K(i)$ is the set of duties covering trip $i \in N$, $K(i,j)$ denotes the set of deadhead tasks corresponding to deadhead $(i,j) \in A^s$ and $K(i,t)$ and $K(s,j)$ denote the deadhead task from the end location of trip i, to the depot and from the depot to the start location of trip j, respectively. Decision variables y_{ij} and x_k are defined as before, that is, y_{ij} indicates whether a vehicle covers trip j directly after trip i or not, while x_k indicates whether duty k is selected in the solution or not. The VCSP can be formulated as follows:

$$\min \sum_{(i,j)\in A} c_{ij} y_{ij} + \sum_{k\in K} d_k x_k \tag{9}$$

$$\sum_{\{j:(i,j)\in A\}} y_{ij} = 1 \qquad \forall i \in N, \tag{10}$$

$$\sum_{\{i:(i,j)\in A\}} y_{ij} = 1 \qquad \forall j \in N, \tag{11}$$

$$\sum_{k\in K(i)} x_k = 1 \qquad \forall i \in N, \tag{12}$$

$$\sum_{k\in K(i,j)} x_k - y_{ij} = 0 \qquad \forall (i,j) \in A^s, \tag{13}$$

$$\sum_{k\in K(i,t)} x_k - y_{it} - \sum_{\{j:(i,j)\in A^l\}} y_{ij} = 0 \qquad \forall i \in N, \tag{14}$$

$$\sum_{k\in K(s,j)} x_k - y_{sj} - \sum_{\{i:(i,j)\in A^l\}} y_{ij} = 0 \qquad \forall j \in N, \tag{15}$$

$$x_k, y_{ij} \in \{0,1\} \qquad \forall k \in K, \forall (i,j) \in A. \tag{16}$$

As before, the objective coefficients c_{ij} and d_k denote the vehicle cost of arc $(i,j) \in A$, and the crew cost of duty $k \in K$, respectively. The objective is to minimize the sum of total vehicle and crew costs. The first two sets of constraints (10) and (11) are equivalent to the quasi-assignment part of formulation for the VSP discussed in Subsect. 3.1. Constraints (12) assure that each trip i will be covered by one duty in the set $K(i)$. Furthermore, constraints (13), (14) and (15) guarantee the link between deadhead tasks and deadheads in the solution, where deadheads corresponding to short and long arcs in A are considered separately. In particular, constraints (13) guarantee that each deadhead from i to j is covered by a duty in the set $K(i,j)$. The other two constraint sets (14) and (15) ensure that the deadheads from the end location of trip i to t and from s to the start location of trip j, possibly corresponding to long arc $(i,j) \in A$, are both covered by one duty. Note that the structure of these last three sets of constraints is such that each constraint corresponds to the selection of a duty from a large set of duties, if the corresponding y variable has value 1. The model contains $|A| + |K|$ variables and $5\,|N| + |A^s|$ constraints, which may already be quite large for instances with a small number of trips.

The difference between the variants with respect to changeovers is in the definition of K. If changeovers are not allowed, there are less possible duties than if changeovers are allowed.

4.2 General Solution Method VCSP

In Fig. 4 we give an outline of our algorithm to compute a lower bound on the optimal solution and to get feasible solutions.

Step 0: Initialization
Solve VSP and CSP (using the algorithm of Fig. 3) and
take as initial set of columns the duties in the CSP-solution.
Step 1: Computation of lower bound
Solve a Lagrangian dual problem with the current set of columns.
Step 2: 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 3;
otherwise, return to Step 1.
Step 3: Construction of feasible solutions
Based on feasible vehicle solutions from Step 1, construct
corresponding feasible crew solutions using the algorithm of Fig. 3.

Fig. 4. Solution method for VCSP

We compute the lower bound in Step 1 by first replacing the equality signs in constraints (12 - 15) by greater-than-or-equal signs, and then applying Lagrangian relaxation with respect to these constraints. To compute the best Lagrangian lower bound we apply subgradient optimization. In every iteration of the subgradient optimization we get a feasible vehicle schedule and we use these schedules to construct feasible (vehicle and crew) solutions at the end. For further details about the Lagrangian relaxation we refer again to Freling et al. (2000).

The main difference to the CSP is that we cannot generate the total set of duties before we start with the optimization algorithm, because we do not have a vehicle solution in advance. So we have to generate the duties during the process, where we have to take into account that the round-trip-condition can be violated at most once in every duty. For this reason we have to construct pieces with no violation and pieces with one violation. How this is done, is described in Subsect. 4.3 and 4.4. After that we just combine pairs of pieces into duties.

In every iteration i we compute an estimate LBT_i of the lower bound for the overall problem. Let LBS_i denote the value of the Lagrangian lower bound in iteration i, then the estimate is computed as:

$$LBT_i = LBS_i + \sum_{k \in K_i} \overline{d}_k \qquad (17)$$

where K_i is the set of duties added in iteration i.

LBT_i is a lower bound for the overall problem if all duties with negative reduced costs are added to the master problem. Of course, we can stop if LBT_i is equal to LBS_i, but in practice we stop earlier, namely if the relative difference is small or if there is no significant improvement in LBS_i during a number of iterations.

At the end we compute feasible crew schedules by solving instances of the CSP that are defined by the (feasible) vehicle schedules from the last iterations. It is also possible to do another subgradient optimization after the convergence and compute crew schedules based on a few vehicle schedules from the last iterations of this optimization.

4.3 Generation of Pieces Without Violation of the Round-trip-condition

Recall that if we want to generate duties with at most one violation of the round-trip-condition, we have to generate pieces with no violation and pieces with at most one violation. In this subsection we look at pieces where the round-trip-condition is not violated and in the next subsection we look at pieces where the round-trip-condition can be violated at most once.

Network Structure

We generate pieces using a network that is an extension of the quasi-assignment network G for vehicle scheduling (see Subsect. 3.1). Let a *start point* (*end point*) be defined as the point corresponding to the start (end) of a vehicle trip. We define the acyclic network $G^p = (N^p, A^p)$, where nodes correspond to the start and end points of each trip, and a source s and a sink t representing the depot. Arcs in A^p correspond to trips, deadheads and layovers. Fig. 5 illustrates network G^p with six trips, six deadheads from and six to the depot and eight layovers. The trips (2,3,4) and (4,5,6) are combinations that violate the round-trip-condition.

The cost associated with an arc $(i,j) \in A^p$ is equal to the value of the Lagrangian multiplier corresponding to this arc. Every arc corresponds to one of the relaxed restrictions in the model VCSP in Subsect. 4.1. Thus, the costs on the arcs are defined such that the cost of a path corresponds to the reduced cost of a piece. Each path $P(u,v)$ between two nodes u and v on network G^p corresponds to a feasible piece of work if its duration is between the minimum

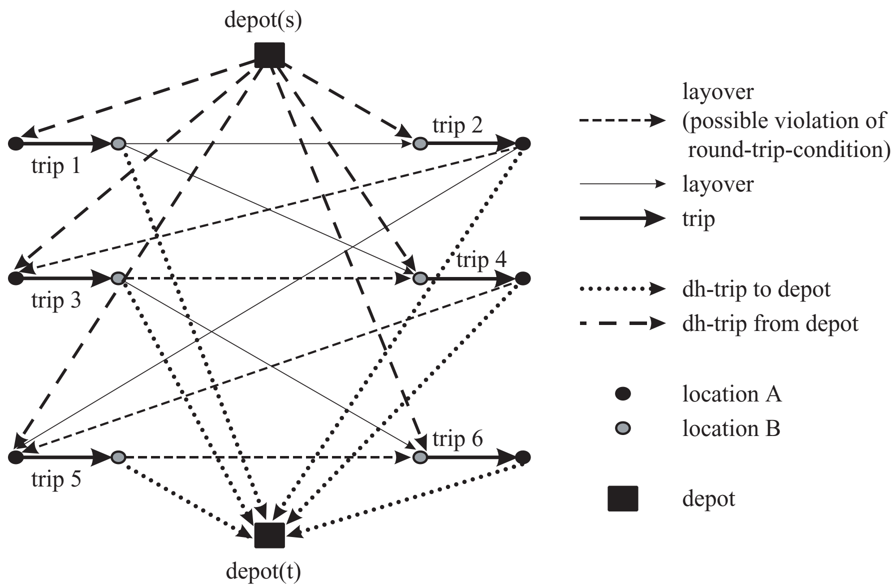

Fig. 5. Piece network

and maximum allowed duration of a piece of work. The duration of a piece of work starting at s and/or ending at t is determined by incorporating travel, sign-on, sign-off and layover times.

We do not need to generate all feasible pieces, but only pieces that correspond to a shortest path between every two pair of nodes. In Freling et al. (2000) we prove that if no duties with negative reduced cost can be constructed from these pieces, then the relaxation has been solved to optimality. If we would not have the problem of the round-trip-condition, finding all shortest paths is very easy (see Freling et al. (2000)), but in our case solving this restricted shortest path problem is not very straightforward. We have considered two ways of solving this problem, namely by an exact procedure or by a fast heuristic.

Exact Shortest Path Algorithm

Because we want to generate pieces where the round-trip-condition is never violated, we define a new acyclic network $G^{p2} = (N^{p2}, A^{p2})$. The set of nodes consists of two nodes for every trip (one which we can reach from the depot and one where we can go to the depot) and we have a node for every deadhead in A^p. The source and the sink correspond again to the depot and the set of arcs is now defined by deadheads from and to the depot and a combination of a trip and a layover or only a trip. For the same example as in Fig. 5 we have the network G^{p2} in Fig. 6.

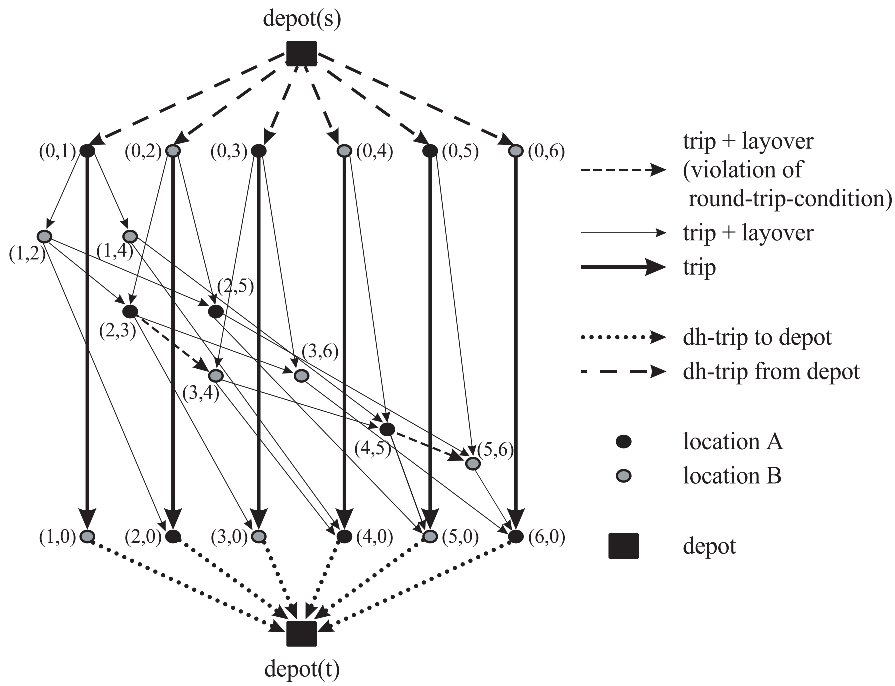

Fig. 6. Better representation of piece network

We can generate the pieces now by just deleting the arcs corresponding to violations of the round-trip-condition (in the example the arcs from (2,3) to (3,4) and from (4,5) to (5,6)) and solve the shortest path problem between every node u of type (0,..) and every possible end node v. For all feasible paths from u to v, three additional paths are considered, namely path $s, u, ..., v$, path $u, ..., v, t$ and path $s, u, ..., v, t$.

Heuristic for Generating Pieces

Because we have to generate the pieces many times, we have developed a heuristic for generating the pieces. We do this by solving two "shortest" path problems in the network G^p between every pair of nodes. We explain this heuristic via the example in Fig. 5. For solving the first "shortest" path problem we delete the arcs from 2 to 3 and from 4 to 5 (the parts of the violation of the round-trip-condition at location A) and compute after that the shortest path. For the other "shortest" path we delete the arcs from 3 to 4 and from 5 to 6 (parts corresponding to location B). Finally, we choose the best of the two "shortest" paths. In this way, we always get a piece that never violates the round-trip-condition, although it is possible that we do not get the piece with the lowest reduced cost. In the example, the piece with the lowest reduced cost can be a piece consisting of trips 3, 4 and 5, which can never be the solution of our heuristic. However, from computational experiments we have concluded that this heuristic works very well for the problems at the RET.

4.4 Generation of Pieces with at Most One Violation of the Round-trip-condition

In this case, the pieces can have at most one violation of the round-trip-condition. Of course, we can generate here also pieces with no violation, for example, if there is no violation possible or if it is not attractive to violate the round-trip-condition. We can easily adapt the exact algorithm and the heuristic for this case. Previously, we computed the shortest path in an acyclic network from u to v by just computing the shortest path from u to every node u' connected with u and so on. In this case, we have to remember two paths to u', namely the shortest path with at most one violation of the round-trip-condition and the shortest path without violation of the round-trip-condition. We do this for all nodes between u and v such that we can add a violation at the moment we want.

4.5 Relief Restrictions

Recall that the RET has several restrictions with respect to relieving:

1. a duty that does not start at the depot, starts with a relief time and at a relief location;
2. a duty that does not end at the depot, ends after a layover and at a relief location.

These restrictions also hold for the start/end of every piece in the variants where changeovers are not always allowed. The difficulty of these restrictions is that a duty cannot end at every relief location, because the time between the departure and the arrival must be at least the relief time. We solve this problem in the following way: first we construct a set of all possible layovers where a duty cannot end. We can do this in advance and when constructing the pieces we assure that a piece never ends at such a layover. Of course, we also introduce this restriction now for the pieces before the break, although that is not required. This is not a problem, because if a duty ends with a layover before the break, there is another duty with the same properties that ends with a trip.

5 Computational Results on RET Data

This section deals with a computational study of the integration of bus and driver scheduling. We have used data of two individual bus lines from the RET (line 35 and 38). These lines are used to compare the different approaches for bus and driver scheduling, that is, we investigate the effectiveness of integration as compared to the traditional sequential approach and we compare the different variants of allowing changeovers. The following points are relevant for all the different approaches proposed in this section:

1. The objective is to minimize the number of drivers and buses in the schedule, therefore we use fixed costs. At the RET, the fixed cost for a driver is about equal to the fixed cost for two buses. As an indication, we take as fixed cost for the bus 600 cost units and for the driver 1,200 cost units. For a fair comparison between the sequential and integrated approach, we also need variable vehicle costs per minute for every deadhead, because if we do not add variable vehicle costs, the VSP will only minimize the number of buses during peak hours and the number of buses during off-peak hours may be much too large. Because every bus needs a driver, in the off-peak hours the number of drivers is far from optimal. Therefore the total number of drivers is much higher than if we add variable vehicle costs. We take as variable vehicle cost one cost unit per minute.
2. The different variants, as described in Subsect. 2.1, are called a and b, where a is changeovers are only allowed if there is enough time to relief one driver by another and b is the current RET rule that changeovers are allowed for split duties and not for other types of duties.
3. The column generation is terminated once the difference between the real and the current lower bound is less than 0.1% or if no significant improvement in the lower bound is obtained for a certain number (*max_tailing_off*) of iterations; this is called *tailing-off criterion*. We have considered an improvement significant if it was at least 5%.
4. For the integrated approach we generate 10 feasible solutions, where 5 follow from the last iterations of the column generation and the other 5 follow from the last iterations of the last subgradient optimization (see Subsect. 4.2).
5. The strategies for generating columns in the pricing problem are obtained after extensive testing and tuning (see Freling (1997)).
6. All tests are executed on a Pentium II 350 pc with 64Mb of computer memory.

In Table 1 we summarize the size of the problem and the important properties for line 35 and 38. We have only considered the problems for weekdays.

The lines 35 and 38 are representative for the RET, where line 35 is of middle size and line 38 is of large size (e.g. line 38 is the largest line of the RET if we look at the number of trips). In Table 2, we show the results of line 35 and line 38. For these results we have used the heuristic for generation the pieces described in Subsect. 4.3.

Both tables consist of four parts, two for the sequential and two for the integrated approach. In the first part we summarize the results for the lower

	Line 35	Line 38
number of trips	131	259
number of round-trip-condition restrictions	6	39
number of relief restrictions	48	93
types of duties	8	8
number of relief locations	1	1
deadheads between two endlocations allowed	yes	no
max_tailling_off	50	15

Table 1. Important properties of line 35 and 38

		Line 35		Line 38	
		variant a	**variant b**	**variant a**	**variant b**
sequential	lower: buses	10	10	14	14
	lower: drivers	16	17	21	23
	lower: total costs	26634	27834	35653	38053
	number of buses	10	10	14	14
	number of drivers	16	17	23	24
	total costs	26638	27838	38073	39273
integrated	lower: total costs	24945	24968	35553	36040
	iterations	24	20	19	19
	cpu (sec.)	279	232	2433	2502
	number of buses	10	10	13	13
	number of drivers	15	15	22	23
	total costs	25495	25573	36438	37755
	gap (%)	2.16	2.37	-	-

Table 2. Results line 35 and 38

bound phase of the VSP (number of buses), the CSP (number of drivers) and of the total vehicle and crew costs. In the second part we show the characteristics of the feasible solution found by the sequential approach. We do not give the computation times for the sequential approach, because they can be neglected. For the integrated approach we give for line 35 the lower bound after convergence and for line 38 the lower bound of the last iteration. The lower bound phase for line 38 terminated, because the tailing-off criterion is satisfied. We also give the number of column generating iterations and the total computation time for the lower bound phase. In the last part we give the number of buses, drivers and the total costs for the best of the ten feasible solutions. For line 35 we also give the relative gap between the lower and the upper bound. We cannot do this for line 38, because there we do not have a lower bound for the total problem but only for a subproblem, since the algorithm terminated before convergence. This is the reason why we have also computed a “real” lower bound, obtained when the tailing-off criterion was not used. These results are given in Table 3.

		variant a	**variant b**
sequential	lower: buses	14	14
	lower: drivers	21	23
	lower: total costs	35653	38053
	number of buses	14	14
	number of drivers	23	24
	total costs	38073	39273
integrated	lower: total costs	33152	33273
	iterations	105	113
	cpu (hour)	4.7	4.9
	number of buses	13	13
	number of drivers	21	22
	total costs	35040	36311
	gap (%)	5.39	8.37

Table 3. Results line 38 after convergence

In Table 3 one can see that we can even compute for the largest RET problem a “real” lower bound within 5 hours. Furthermore, in both variants the solutions are better than the solution that was obtained when the tailing-off criterion was used.

For all the variants the integrated approach gave better results than the sequential approach and the gap for the integrated approach is always lower than 10%. For example in Table 3, we saved two drivers and one bus (about 8% reduction in costs) for the variants a and b and we still had a gap of 5.39% and 8.37%, respectively. Because of the possibility to violate the round-trip-condition once in every crew duty, we can even save buses in the integrated approach. (Without this possibility only drivers could have been saved.) Another important result is that we find solutions with less drivers if more flexibility for changeovers is allowed. We have to be careful with interpreting these results, however, because the difference in the lower bounds is negligible. Hence, it is possible that our approach simply works better in these variants, while the optimal solutions are actually the same.

In Huisman (1999) one can find more results with some other variants and cost structures. For example, we took different costs for split duties and we added an extra restriction on the maximum length of the break. We also looked at a problem where no changeovers are allowed at all and where changeover are always allowed. For the first situation we applied a different model (see also Freling et al. (2000)).

Exact Computation of the Lower Bound

When interpreting the above results, one should keep in mind that we used a heuristic for generating the pieces, i.e., we did not necessarily take the pieces with the lowest reduced cost. So it is possible that we found that there were no duties left with negative reduced cost with the pieces we generated, whereas a combination of two pieces with negative reduced cost did exist. In that case, we did not find a real lower bound for the total problem, but only an approximation of this lower bound. Therefore we have compared this approximation with the lower bound obtained by using the exact shortest path algorithm, described in Subsect. 4.3, to generate the pieces. This was done for several lines and variants. A typical example of our findings is given in Table 4, which summarizes the results for line 35, variant b.

	Heuristic	**Exact (normal)**	**Exact (equality)**
lower: total costs	24968	24846	25052
iterations	20	40	66
cpu (sec.)	232	2198	4138

Table 4. Comparison of the heuristic and exact method to compute the lower bound

The column “exact (normal)” refers to the situation that the method and all parameters are the same as for the heuristic. This means that the exact lower bound is lower than the lower bound computed with the heuristic, although not much. Moreover, we can obtain a better exact lower bound if we do not compute the Lagrangian relaxation with “greater-than-or-equal-to” signs in constraints (12 - 15), but with equality signs instead. The results of this approach are given in the column “exact (equality)”. We see that the lower bound of the heuristic is actually a real lower bound.

6 Conclusions

In this paper we applied an integrated approach to vehicle and crew scheduling at the RET. We handled complicating constraints, which have not been considered in the literature before. We showed the results for two individual bus lines, including the RET bus line with the largest number of trips. For these lines the integrated problem could be solved in a reasonable amount of time, where the gap between the lower bound and the best feasible solution was less than 10% in all cases. The main conclusion is that we can save vehicles and/or crews by integrating the vehicle and crew scheduling problem, which may lead to a big decrease in costs. Especially, in the case that (almost) no changeovers are allowed, integration is very attractive.

Another important result is that sometimes it is indeed possible to reduce the total costs by allowing changeovers more often.

Acknowledgment From 1996 until 1999, the second author was also affiliated with the RET, the public transport company in Rotterdam, the Netherlands. The authors are grateful to the RET company for providing the data and for supporting the research.

Bibliography

Darby-Dowman, K., J.K. Jachnik, R.L. Lewis, G. Mitra (1988). Integrated decision support systems for urban transport scheduling: discussion of implementation and experience. In: Daduna, J.R., A. Wren (Eds.), *Computer-Aided Transit Scheduling: Proceedings of the Fourth International Workshop*, Berlin: Springer Verlag. pp. 226-239.

Freling, R. (1997). *Models and Techniques for Integrating Vehicle and Crew Scheduling*. Amsterdam: Thesis Publishers.

Freling, R., D. Huisman, A.P.M. Wagelmans (2000). Models and Algorithms for integration of vehicle and crew scheduling. Report EI2000-10/A. Econometric Institute, Erasmus University Rotterdam, Rotterdam, the Netherlands.

Freling, R., J.M. Pinto Paixão, A.P.M. Wagelmans (1995). Models and algorithms for vehicle scheduling. Report 9562/A. Econometric Institute, Erasmus University Rotterdam, Rotterdam, the Netherlands. To appear in *Transportation Science.*

Huisman, D. (1999). *Sequentiële en geïntegreerde benaderingen voor voertuig- en personeelsplanning bij de RET* (in Dutch). Master's thesis. Faculty of Economics, Erasmus University Rotterdam, the Netherlands.

Rousseau, J.M., J.Y. Blais. HASTUS: An interactive system for buses and crew scheduling. In: Rousseau J.M. (Ed.), *Computer Scheduling of Public Transport 2*, Amsterdam: North Holland. pp. 473-491.

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

REPORT SERIES
RESEARCH IN MANAGEMENT

Publications in the Report Series Research* in Management

Impact of the Employee Communication and Perceived External Prestige on Organizational Identification
Ale Smidts, Cees B.M. van Riel & Ad Th.H. Pruyn
ERS-2000-01-MKT

Critical Complexities, from marginal paradigms to learning networks
Slawomir Magala
ERS-2000-02-ORG

Forecasting Market Shares from Models for Sales
Dennis Fok & Philip Hans Franses
ERS-2000-03-MKT

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

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

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

Start-Up Capital: Differences Between Male and Female Entrepreneurs, 'Does Gender Matter?'
Ingrid Verheul & Roy Thurik
ERS-2000-07-STR

The Effect of Relational Constructs on Relationship Performance: Does Duration Matter?
Peter C. Verhoef, Philip Hans Franses & Janny C. Hoekstra
ERS-2000-08-MKT

Marketing Cooperatives and Financial Structure: a Transaction Costs Economics Analysis
George W.J. Hendrikse & Cees P. Veerman
ERS-2000-09-ORG

---

* ERIM Research Programs:
LIS Business Processes, Logistics and Information Systems
ORG Organizing for Performance
MKT Decision Making in Marketing Management
F&A Financial Decision Making and Accounting
STR Strategic Renewal and the Dynamics of Firms, Networks and Industries

A Marketing Co-operative as a System of Attributes: A case study of VTN/The Greenery International BV,
Jos Bijman, George Hendrikse & Cees Veerman
ERS-2000-10-ORG

Evaluating Style Analysis
Frans A. De Roon, Theo E. Nijman & Jenke R. Ter Horst
ERS-2000-11-F&A

From Skews to a Skewed-t: Modelling option-implied returns by a skewed Student-t
Cyriel de Jong & Ronald Huisman
ERS-2000-12-F&A

Marketing Co-operatives: An Incomplete Contracting Perspective
George W.J. Hendrikse & Cees P. Veerman
ERS-2000-13– ORG

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

Ownership Structure in Agrifood Chains: The Marketing Cooperative
George W.J. Hendrikse & W.J.J. (Jos) Bijman
ERS-2000-15-ORG

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

Organizational Change and Vested Interests
George W.J. Hendrikse
ERS-2000-17-ORG

Strategies, Uncertainty and Performance of Small Business Startups
Marco van Gelderen, Michael Frese & Roy Thurik
ERS-2000-18-STR

Creation of Managerial Capabilities through Managerial Knowledge Integration: a Competence-Based Perspective
Frans A.J. van den Bosch & Raymond van Wijk
ERS-2000-19-STR

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

Currency Hedging for International Stock Portfolios: A General Approach
Frans A. de Roon, Theo E. Nijman & Bas J.M. Werker
ERS-2000-21-F&A

Transition Processes towards Internal Networks: Differential Paces of Change and Effects on Knowledge Flows at Rabobank Group
Raymond A. van Wijk & Frans A.J. van den Bosch
ERS-2000-22-STR

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

Creating the N-Form Corporation as a Managerial Competence
Raymond vanWijk & Frans A.J. van den Bosch
ERS-2000-24-STR

Competition and Market Dynamics on the Russian Deposits Market
Piet-Hein Admiraal & Martin A. Carree
ERS-2000-25-STR

Interest and Hazard Rates of Russian Saving Banks
Martin A. Carree
ERS-2000-26-STR

The Evolution of the Russian Saving Bank Sector during the Transition Era
Martin A. Carree
ERS-2000-27-STR

Is Polder-Type Governance Good for You? Laissez-Faire Intervention, Wage Restraint, And Dutch Steel
Hans Schenk
ERS-2000-28-ORG

Foundations of a Theory of Social Forms
László Pólos, Michael T. Hannan & Glenn R. Carroll
ERS-2000-29-ORG

Reasoning with partial Knowledge
László Pólos & Michael T. Hannan
ERS-2000-30-ORG