Available online at www.sciencedirect.com

ScienceDirect

European Journal of Operational Research 191 (2008) 19–31

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/ejor

Discrete Optimization

Combined vehicle routing and scheduling with temporal precedence and synchronization constraints

David Bredström, Mikael Rönnqvist *

Norwegian School of Economics and Business Administration, Bergen, Norway

Received 3 November 2006; accepted 15 July 2007
Available online 1 November 2007

Abstract

We present a mathematical programming model for the combined vehicle routing and scheduling problem with time windows and additional temporal constraints. The temporal constraints allow for imposing pairwise synchronization and pairwise temporal precedence between customer visits, independently of the vehicles. We describe some real world problems where in the literature the temporal constraints are usually remarkably simplified in the solution process, even though these constraints may significantly improve the solution quality and/or usefulness. We also propose an optimization based heuristic to solve real size instances. The results of numerical experiments substantiate the importance of the temporal constraints in the solution approach. We also make a computational study by comparing a direct use of a commercial solver against the proposed heuristic, where the latter approach can find high quality solutions within specific time limits.

Keywords: Routing; Scheduling; Temporal constraints; Synchronization; Branch and bound

1. Introduction

Combined vehicle routing and scheduling with time windows arises in many applications and there is an extensive and wide research literature on operations research (OR) models and methods, both exact and heuristic. Temporal constraints within a route for one vehicle frequently occurs in well known problems such as the dial-a-ride and the pickup and delivery problems. However, the problem of vehicle dependencies is given much less attention in the literature. A typical application is when two vehicles must meet at a point at the same time or when a vehicle cannot pick up a load until another vehicle has delivered the same load. The main goal of this paper is to develop and test a general mathematical programming model for the combined vehicle routing and scheduling with time windows and additional temporal constraints. The temporal constraints introduced allow for imposing pairwise temporal precedence and pairwise synchronization between customer visits, independently of the vehicles.

Given a fleet of vehicles available in a depot and a set of customers to be serviced within their respective prescribed time windows, the objective for the vehicle routing and scheduling problem (VRSP-TW) is

* Corresponding author.
E-mail address: mikael.ronnqvist@nhh.no (M. Rönnqvist).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2007.07.033

for example, to minimize the total traveling time. Both heuristic and exact solution methods have been suggested for solving applications of the VRSP-TW, see, e.g. the survey in Desrosiers et al. (1995). The VRSP with a single vehicle and precedence constraints is commonly seen as a traveling salesman problem with precedence constraints. Fagerholt and Christiansen (2000) use the single vehicle VRSP-TW with additional allocation constraints to solve a subproblem arising in a ship scheduling application. If we introduce capacity constraints to the VRSP-TW, depending on the precedence constraints, we get a pickup and delivery problem with time windows (PDP-TW) which is an exhaustively studied problem, see, e.g. Desrosiers et al. (1995). Sigurd et al. (2004) use precedence constraints for an application that arise in live animal transportation.

In the pickup and delivery and the dial-a-ride problems, the precedence constraints are limited to precedence within a route for a single vehicle. A related problem is the job shop scheduling problem (JSP), where each job is defined by a set of ordered activities and each activity is normally to be executed on one predefined resource. All activities for one job are not bound to one resource and the precedence constraints therefore span over multiple resources, as opposed to the pickup and delivery and the dial-a-ride problems. Beck et al. (2003) study the differences between VRP and JSP and apply both vehicle routing and scheduling techniques to VRPs. In their study, they include vehicle independent precedence constraints to the VRP and observe that the routing techniques they use have difficulties in finding feasible solutions, while the scheduling techniques find feasible solutions to all the studied problem instances. Gélinas and Soumis (2005) use decomposition techniques to solve large scale JSP problems. They obtain tighter LP relaxation by using TSP-TW as subproblem for the schedule on one machine instead of the SPP-TW used for VRP-TW problems.

In the combined vehicle and crew scheduling problem for urban mass transit systems, drivers are allowed to change bus in so called relief points. Commonly, as seen in Haase et al. (2001) and the work of Freling et al. (2003), the arrival time at a relief point is defined by a timetable and therefore the synchronized arrival of bus drivers is implicitly considered. In the homecare scheduling problem presented in Eveborn et al. (2006), there is a required synchronization of staff visits to customers (such as elderly people). The model for the periodic routing and airline fleet assignment problem, presented in the paper by Ioachim et al. (1999), has temporal constraints that define the same departure time for pairs of flights, which is a set of synchronization constraints in the sense that we use in this paper. For their problem they develop a multi-commodity flow formulation and a solution method based on a side constrained set partitioning reformulation, which they solve with column generation in a branch-and-bound framework. The solution process is further developed in Bélanger et al. (2006) where characteristics of the subproblem are used.

In this paper, we want to emphasize the importance of the temporal synchronization and precedence constraints found in several real world applications. For this purpose, we suggest a straight-forward model of the VRSP-TW and extend it with the introduced constraints. The main contributions of the paper are as follows. The proposed model is a generalization of the VRSP-TW. Using standard VRSP-TW some strict simplification of the problem must be enforced to handle the synchronization constraints. A standard approach is to put a strict limit on the time windows providing a simplified VRP problem. A model that considers some synchronization constraints for an airline fleet assignment and routing is given in Ioachim et al. (1999). Our model however, is more general and is based on an extension of a traditional VRP model. We also demonstrate through the computational experiments that our proposed model is not significantly harder to solve compared to a VRSP-TW model without synchronization constraints. We also demonstrate the potential improvements in handling the constraints explicitly in the model. We propose an optimization based heuristic that finds high quality solutions within specific time limits. We do not suggest that this model should be used directly to solve all applications. It does however, describe the temporal constraints clearly. It can also be used as a basis for formulating and developing more application oriented models and solution methods.

The outline of this paper is as follows. In Section 2 we describe the problem and demonstrate it with an illustrative example. In this section we also describe some typical applications where precedence and synchronization constraints are important aspects. In Section 3 we present the new model. We focus on constraints that relate to at least two vehicles.

We do not study the case when two jobs are done by the same vehicle such as in dial-a-ride and pickup and delivery. This is easy to include but does not add anything to the model. In Section 4 we describe the numerical tests performed. This includes a description of the test problems, a description of the heuristic developed and analyses of the tests. Finally, we make some concluding remarks and outline some further work in Section 5.

2. Problem formulation

We assume that we have a fleet of vehicles available in a depot, and a set of visits to customers to be serviced within their respective prescribed time window. Let K denote a set of vehicles and let $G = (\overline{N}, A)$ be a directed graph, where $\overline{N} = \{o, d, 1, \ldots, n\}$ is the node set and $A = \{(i,j) | i \neq j, i \in \overline{N} \setminus \{d\}, j \in \overline{N} \setminus \{o\}\}$ is the arc set. The nodes o and d both represent the depot and the nodes $N = \{1, \ldots, n\}$ the visits to customers. Each visit $i \in N$ has an associated time window $[a_i, b_i]$ for the arrival time, and a duration D_i for the visit and for $i \in \{o, d\}$ the time windows $[a_i^k, b_i^k]$ define the availability of the vehicle $k \in K$. For an arc $(i,j) \in A$ we define the traveling time with T_{ij}.

We denote the set of pairwise synchronized visits with $P^{\text{sync}} \subset N \times N$, and the nodes with pairwise precedence constraints with $P^{\text{prec}} \subset N \times N$. For each pair $(i,j) \in P^{\text{prec}}$ we define a temporal offset S_{ij} if it is required that j is visited at least S_{ij} time units after i.

2.1. Example

In the example, we use a network with ten physical customers and where five customers need service from two simultaneous vehicles. We have $N = \{1, \ldots, 15\}$ where $11, \ldots, 15$ are synchronized visits in the pairs $(1, 11), \ldots, (5, 15) \in P^{\text{sync}}$. In Fig. 1a, we show the network with ten nodes where the synchronized visits are indicated with a double circle. The service durations are shown in Fig. 1b where we assume that the durations for both vehicles in a synchronized visit are equal. To show the proportions of traveling time and duration we show some of the traveling time on arcs.

We illustrate the optimal solutions for two problems solved on the network in Fig. 2, where in both problems the vehicles have an availability of 9 hours and the visit time windows are disregarded. In the first example, as seen in Fig. 2a with the schedules

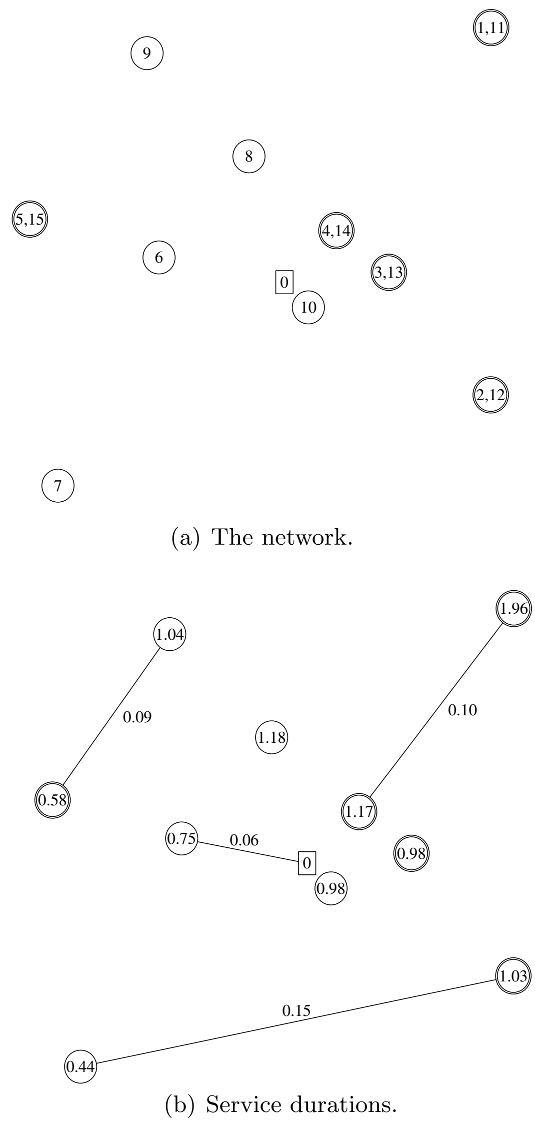
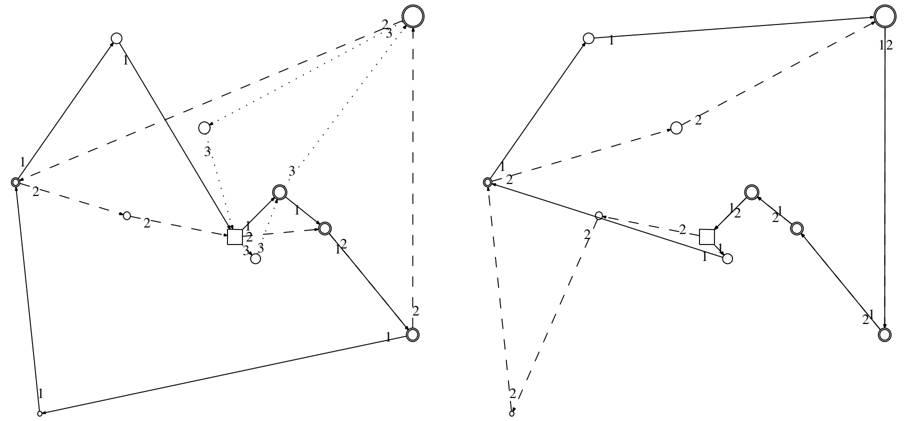


Fig. 1. Example network and durations.

in Table 1, we use three vehicles and our objective is to minimize the sum of the traveling distance and the maximal difference in workload. The difference in workload is measured with the pairwise vehicle difference sum of service durations. In the second example there are two vehicles and the objective is to minimize traveling time, see Fig. 2b and the schedule in Table 2.

In the examples, the service durations occupy a major part of the vehicles' availability time and traveling times are considerably low in comparison. With three vehicles, the total waiting time is 9.62 hours while the solution to the problem with two vehicles has a total waiting time of 0.68 hour.



(a) The optimal solution when minimizing the sum of traveling distance and work load for 3 vehicles.

(b) The optimal solution when minimizing the traveling distance for 2 vehicles.

Fig. 2. Example solutions to two problems on one network.

Table 1
Schedules for the three vehicles in the first example

	Arrival	Duration	Traveling	Waiting
Vehicle 1				
Depot	0.00	0.00	0.04	0.99
4, 14	1.03	1.17	0.04	0.00
3, 13	2.24	0.98	0.06	0.00
2, 12	3.28	1.03	0.15	2.14
7	6.60	0.44	0.13	0.00
5, 15	7.17	0.58	0.09	0.00
9	7.84	1.04	0.11	0.00
Depot	9.00	0.00		
	Sum	5.25	0.62	3.13
Vehicle 2				
Depot	0.00	0.00	0.05	2.19
3, 13	2.24	0.98	0.06	0.00
2, 12	3.28	1.03	0.14	0.55
1, 11	5.00	1.96	0.22	0.00
5, 15	7.17	0.58	0.06	0.37
6	8.19	0.75	0.06	0.00
Depot	9.00	0.00		
	Sum	5.30	0.60	3.10
Vehicle 3				
Depot	0.00	0.00	0.01	0.00
10	0.01	0.98	0.04	0.00
4, 14	1.03	1.17	0.10	2.69
1, 11	5.00	1.96	0.11	0.70
8	7.77	1.18	0.05	0.00
Depot	9.00	0.00		
	Sum	5.29	0.92	3.39

The maximal differences in total duration for the vehicles is only 0.05 hour in the first example.

Table 2
Schedules for the two vehicles in the second example

	Arrival	Duration	Traveling	Waiting
Vehicle 1				
Depot	0.00	0.00	0.01	0.00
10	0.01	0.98	0.13	0.41
5, 15	1.53	0.58	0.09	0.11
9	2.31	1.04	0.15	0.00
1, 11	3.51	1.96	0.14	0.00
2, 12	5.61	1.03	0.06	0.08
3, 13	6.78	0.98	0.04	0.00
4, 14	7.79	1.17	0.04	0.00
Depot	9.00	0.00		
	Sum	7.74	0.66	0.60
Vehicle 2				
Depot	0.00	0.00	0.06	0.00
6	0.06	0.75	0.14	0.00
7	0.95	0.44	0.13	0.00
5, 15	1.53	0.58	0.10	0.00
8	2.21	1.18	0.11	0.00
1, 11	3.51	1.96	0.14	0.00
2, 12	5.61	1.03	0.06	0.08
3, 13	6.78	0.98	0.04	0.00
4, 14	7.79	1.17	0.04	0.00
Depot	9.00	0.00		
	Sum	8.10	0.82	0.08

2.2. Applications

To illustrate the practical importance of the temporal precedence and synchronization constraints, we describe a limited number of applications. Each of these, has a different set of temporal constraints. How these are included (or simplified) in the planning and solution process differs and we describe some approaches. In the description, we focus on how the new synchronization constraints are handled. We start by describing a homecare staff scheduling problem and then move to two forest operations problems. We describe the homecare problem in more detail because it forms the base for the numerical experiments.

2.3. Homecare staff scheduling

Daily planning of homecare staff is a combined scheduling and routing problem. The planning problem is to establish routes for staff members where each route includes a set of visits. A description of the problem, solution methods and a decision support system is given in Eveborn et al. (2006). The problem is solved every day depending on the actual situation and taking into account, e.g. sick staff and new or changed visits. Each staff member has a set of skills, e.g. language skill, medical certificate and gender. Each member also has particular working hours, e.g. half time or full time. All staff members usually begin and end the day in one base position where the final alterations to plans are made and where visit reports are handed in after completing the routes.

Each visit to a customer has a given time window and duration and requires particular skills. A customer is typically an elderly person living at home and who has an agreement with the local community about support, e.g. for cleaning, bathing, cooking or medical attention. If the visit is for cooking and for perhaps 1 hour, the time window may be limited so that the visit starts at between 10.30 and 11.30 a.m. If the visit is for cleaning, the start may be anywhere between 8.00 a.m. and 3.00 p.m. The goal is to find the best possible plan. For each combination of staff member and task, there is a bonus number associated. The bonus is calculated from the preferences on visits of the staff members and quality measures at the customers and is expressed with an estimated artificial value which is weighted against the total traveling time. A typical quality measure is that the same staff member returns to the same customer (seen over many planning periods).

Each staff member follows his or her own route. However, some customers require that two staff members are present simultaneously or in a given order. This is, for example, due to heavy lifts while bathing a customer. An example of a precedence constraint is when medication has to be given by a qualified nurse before or after food. Typically, five to ten percent of the tasks follow such requirements. This creates a difficult problem in the planning as the solution methods are based on VRP heuristics and cannot include the synchronization between two staff members (or vehicles in the VRP model). The solution approach for this is to use two synchronized customer visits. Each must be made by a staff member and in order to do this at the same time, the time window for the start of the visit is set such that there is only one fixed starting time (i.e. no flexibility). This works in practice, but it is difficult to decide the schedule of these visits. It may lead to a situation where no feasible plan is found among the given staff and therefore extra staff has to be called in.

There are many staff planning problems that are similar to the home care problem described above. One such application is the planning of security guards. Visits in this context are to a location, e.g., an industrial site to check doors, windows and locks. Each guard has a skill depending on, e.g. if she or he has a dog or is trained for specific tasks at customers. The synchronization constraints arise when there is a need for more than one guard to make a visit. Precedence constraints arise when there is a need to make, perhaps, three different and ordered visits to the same location but when there is a freedom when they are done.

2.4. Forest operations

In forest management, harvesting and truck routing are two important operations. Harvesting operations at harvest areas (or stands) are done by two types of vehicles: harvesters and forwarders. Harvesters fell the trees and cut them into logs that are put in piles in the stand. Later, forwarders come along and pick up the piles and move them to larger piles adjacent to forest roads where logging trucks pick them up for further transportation to mills and terminals. Harvest planning is typically done on an annual basis and includes description of stands and demands at mills for example. A description of the problem, models and methods is provided in Karlsson et al. (2004). One result is a monthly allocation of stands to be harvested throughout one year. Once this annual planning is done, there is a need to decide the routes for a number of harvesters and forwarders active in the planning district. Each vehicle has a given size and capacity and needs a particular time to perform the operations at each stand. The time and cost to move a vehicle depends on the distance between stands. If the distance is short, typically less than 5 km, the vehicle drives by itself. Otherwise it is lifted to a trailer and moved. The ideal situation would be to plan individual routes for the harvesters and forwarders so that their capacity can be utilized efficiently. One precedence restriction is that forwarding can be done once the harvesting is done. Another is that the forwarding should be done within a specified time after the harvesting. Because of this, traditional VRP methods cannot be applied directly. The approach taken is therefore to turn harvesters and forwarders into teams. A team consists of a harvester and forwarder of similar size. In this way, both operations can be done by a team and the problem becomes a standard VRP problem. However, as the vehicles within a team need different times for a stand, performance is limited to the slowest.

The routing of logging trucks is done on a daily or weekly basis. The underlying routing of trucks is a pick up and delivery problem. One or several piles of a given assortment (or product) is picked up and then delivered to a delivery point. There are two principal types of trucks: with or without a crane. Trucks with cranes can load and unload themselves and trucks without depend on loaders at stands and delivery points. The reason for not having a crane is that without a crane there is an increased weight capacity, typically two metric tons, which relates to about 5–10%. Loaders are only located to areas where there is a certain level of harvesting. These loaders often serve several harvest areas and move between these in order to load trucks. To find an optimal plan, the routes for the trucks need to be planned together with the routes of the loaders. One synchronization constraint is that a loader must be at a harvest are when the truck arrives. In practice, the loaders are planned separately. Given the schedule of the loaders, a set of time windows when the harvest areas are "open" for non-crane trucks is determined. This is then used in solving the pick up and delivery problem for the logging trucks. A description of models and methods for the routing of logging trucks can be found in Palmgren et al. (2003).

3. Mixed integer programming model

In this section, we present a mixed integer programming formulating of the problem where we use two types of variables: the routing variables $x_{ijk} \in \{0, 1\}$ and the scheduling variables $t_{ik} \geqslant 0$. The routing variable x_{ijk} is one if the vehicle $k \in K$ traverse the arc $(i, j) \in A$. For simplicity we use one set of arcs A defined for all vehicles. In applications where some customers can only be served by some vehicles, there are vehicle dependent networks. The scheduling variable t_{ik} is the time when the vehicle k arrives at the visit $i \in N$ and is zero if the vehicle k does not visit i. The routing and scheduling constraints are modeled as follows:

$$\sum_{k\in K}\sum_{j:(i,j)\in A} x_{ijk} = 1 \quad \forall i \in N, \tag{1}$$

$$\sum_{j:(o,j)\in A} x_{ojk} = \sum_{j:(j,d)\in A} x_{jdk} = 1 \quad \forall k \in K, \tag{2}$$

$$\sum_{j:(i,j)\in A} x_{ijk} - \sum_{j:(j,i)\in A} x_{jik} = 0 \quad \forall i \in N \;\; \forall k \in K, \tag{3}$$

$$t_{ik} + (T_{ij} + D_i)x_{ijk} \leqslant t_{jk} + b_i(1 - x_{ijk}) \quad \forall k \in K \quad \forall (i,j) \in A, \tag{4}$$

$$a_i \sum_{j:(i,j)\in A} x_{ijk} \leqslant t_{ik} \leqslant b_i \sum_{j:(i,j)\in A} x_{ijk} \quad \forall k \in K \;\; \forall i \in N, \tag{5}$$

$$a_i^k \leqslant t_{ik} \leqslant b_i^k \quad \forall k \in K \;\; \forall i \in \{o, d\}. \tag{6}$$

Constraints (1)–(6) form the constraint set for a multiple traveling salesman problem, where, if we use the vocabulary of the VRSP, the constraints (1) ensure that each customer visit is visited by exactly one vehicle, (2) and (3) define the routing network, and the constraints (4)–(6) are the scheduling constraints. Constraint (5) implies that $t_{ik} = 0$ if visit i is not visited by the vehicle k. Therefore the arrival time for a visit i is defined by $\sum_{k\in K} t_{ik}$. We use this property to formulate the temporal constraints as follows:

$$\sum_{k\in K} t_{ik} = \sum_{k\in K} t_{jk} \quad \forall (i,j) \in P^{\text{sync}}, \tag{7}$$

$$\sum_{k\in K} t_{ik} \leqslant S_{ij} + \sum_{k\in K} t_{jk} \quad \forall (i,j) \in P^{\text{prec}}. \tag{8}$$

Constraints (7) ensure that the vehicles that visit i and j for $(i, j) \in P^{\text{sync}}$ arrive simultaneously. In our formulation, any waiting will take place at the previous visit (or at the depot). The visit j in a pair (i, j) is typically a synchronized visit representing the need for simultaneous service from a second vehicle to a customer. Universally, we can model the demand of s vehicles for one customer by introducing $s - 1$ visits $i_2, \ldots, i_s$ and the relations (i_1, i_2), $(i_1, i_3), \ldots, (i_1, i_s) \in P^{\text{sync}}$.

By using the temporal precedence constraints in (8) we are able to model several real world situations.

One frequent situation is when we want to ensure that a vehicle does not arrive at one visit j before the service of another specific visit i has been completed. This requirement is modeled by letting $S_{ij} = -D_i$. Another situation is when we have a demand for a second vehicle to arrive while a customer still is being served. For example, in the homecare application, this is the situation when one staff member is visiting a customer and at some point needs assistance with heavy lifts. Assuming that the assisting vehicle (j) can arrive at any time during the visit (i) of the first vehicle, this situation is modeled with $S_{ij} = 0$ and $S_{ji} = D_i$ with (i, j), $(j, i) \in P^{\text{prec}}$. It is worth noting that a synchronization constraint can be formulated with two temporal precedence constraints without offsets. We choose to explicitly formulate the synchronization constraints because of their practical importance.

If we relax the constraints in (1), and disregard the synchronization and vehicle independent precedence constraints (7) and (8), the problem decomposes into one problem for each vehicle and this is used in many heuristic and exact solution methods. Constraints (1), (7) and (8) are usually referred to as complicating constraints, because of the coupling of two sets of otherwise independent variables. The increased complexity imposed by the complication constraints is one reason why in many applications we prefer to avoid the vehicle independent temporal constraints.

One example of constraints that we include for use in the numerical experiments, are the balancing constraints, defined in (9)

$$\sum_{(i,j)\in A} W_{ijk_1} x_{ijk_1} - \sum_{(i,j)\in A} W_{ijk_2} x_{ijk_2} \leqslant w \quad \forall k_1 \in K \;\; \forall k_2 \in K \setminus \{k_1\}. \tag{9}$$

The balancing variable w is defined as the upper bound for the pairwise maximal difference between two vehicles in a weighted arc measure. With $W_{ijk} = D_i$, the measure is in service duration and with $W_{ijk} = T_{ij}$ in traveling time. If the objective is to minimize w, we minimize the maximal difference in the given measure. For example, a fairness measure in the homecare staff scheduling problem may be a measure of workload, and one goal is to minimize the maximal difference in workload for staff members.

If we use an objective function with a weighted sum of preferences, traveling time and one balancing variable, the MIP problem is

$$\min \quad \alpha_P \sum_{k\in K} \sum_{(i,j)\in A} c_{ik} x_{ijk} + \alpha_T \sum_{k\in K} \sum_{(i,j)\in A} T_{ij} x_{ijk} + \alpha_B w, \tag{10}$$

$$\text{s.t.} \quad (1)\text{–}(9), \tag{11}$$

where α_P, α_T and α_B are weights and c_{ik} is a preference measure for vehicle k to serve a customer with visit i.

The model presented here is based on having a fixed fleet of vehicles available in a fixed time window and the demand is to serve all customers. One can consider many important varieties. We can have requirements that the customers for visits p and q have to be serviced by one vehicle for pairs $(p, q) \in P^{\text{vehicle}}$. The constraints, in the notation used in this paper, could be formulated as in (12).

$$\sum_{(q,j)\in A} x_{qjk} = \sum_{(p,j)\in A} x_{pjk} \quad \forall (q,p) \in P^{\text{vehicle}}, \;\; \forall k \in K. \tag{12}$$

We can also consider an objective which is to minimize waiting time, if for instance we know a penalty for not serving a customer, or when we have an incurred cost from waiting time. Another measure of fairness is to minimize the maximal difference in total working time.

4. Numerical experiments

The aim of the numerical tests is to analyze the behaviour of the model and the usefulness of including the new constraints explicitly in the model. The instances were generated to simulate the homecare staff scheduling problem, and in particular to resemble the problems presented in Eveborn et al. (2006). We have used the AMPL/CPLEX modeling environment for all implementations, using CPLEX version 10 and solved the instances on a 2.67 MHz Xeon processor and with 2 GB RAM available. When we solved some of the larger instances, we found that CPLEX had problems finding solutions in a reasonable time. It is well known that the MIP formulation of a VRSP problem used in this paper has a large integrality GAP and we can not expect to solve our model with the CPLEX solver directly. We therefore introduce an optimization based heuristic approach that we use in the tests. This heuristic can be compared with the local branching heuristic (RDT), presented in Fischetti et al. (2004). We will use the notation OPT when we refer to solving the problem using CPLEX directly.

In this section, we start by describing the test problems and then present the heuristic. Then we test the performance of the heuristic on the smaller instances. We also make some analyses of the model characteristics. Next we go on to solve some larger instances and investigate the impact of synchronization constraints and time window size.

4.1. Test problems

We assumed that there are on average five customers for each staff member to visit, of which a total of ten percent are synchronized visits. We also assumed all staff members to be available throughout the whole planning horizon, which was one day and fixed to 9 hours, and we excluded the scheduling of rests. We generated five small sized instances for benchmarking the heuristic algorithm and five realistic sized instances as presented in Table 3. We use five groups of time windows increasing in size, ranging from fixed (F), small (S), medium (M) to large (L) and no time window restrictions (A), such that each larger time window covers the smaller time window.

We note that the average duration time is largest in instances 4, 8 and 9. This implies that these are most tight and generally more difficult to solve. The instances 1–5 have a magnitude of 1900 variables and 2100 constraints, the instances 6–8 of 27,000 variables and 28,000 constraints, and the largest instances 9–10 of 106,000 variables and 109,000 constraints.

The customer locations were uniformly distributed on a square area with the depot located in the center and the durations were randomized with the normal distribution with the goal of having a mean of 5 hours workload (excluding traveling time) for each staff member. The traveling time and durations were scaled and rounded to be integer valued with the result of a time discretization less than 1 minute. The network for instance 8 is illustrated in Fig. 3.

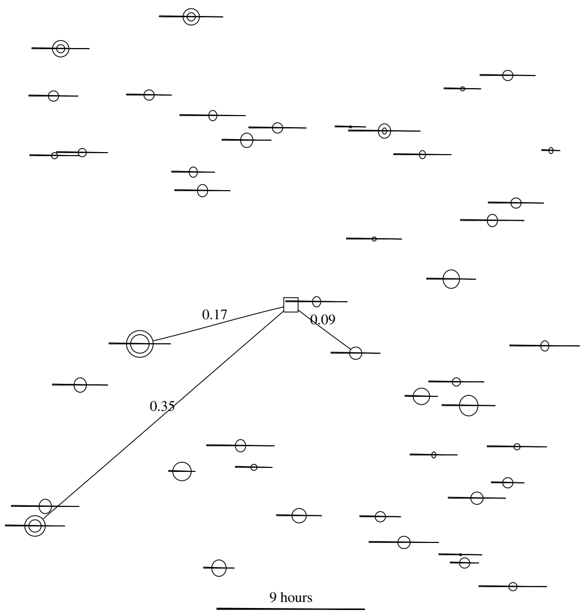


Fig. 3. Instance 8 with time windows (for arrival time) from group L shown proportionally to the durations by the length of the crossing bars. The circle diameter is relative to the visit's duration and the scale relative to 9 hours, printed for reference with the bottom line. The double circles symbolize synchronized visits and the numbers on the arcs the traveling time in hours.

Table 3
Test instances

Instance	$\|N\|$	$\|K\|$	$\|P^{sync}\|$	$\sum D_i/\|K\|$ (hour)	AvTD (hour)	Time windows (hour)				
						F	S	M	L	A
1	20	4	2	4.9	0.22	0	1.5	2.1	2.9	9
2	20	4	2	4.2	0.20	0	1.7	2.2	3.0	9
3	20	4	2	5.3	0.21	0	1.5	2.4	3.0	9
4	20	4	2	5.9	0.29	0	1.8	2.9	3.9	9
5	20	4	2	5.0	0.21	0	1.3	2.1	3.2	9
6	50	10	5	4.7	0.25	0	1.4	2.3	3.1	9
7	50	10	5	5.0	0.23	0	1.6	2.5	3.4	9
8	50	10	5	6.2	0.23	0	1.5	2.4	3.2	9
9	80	16	8	6.1	0.21	0	1.5	2.3	2.9	9
10	80	16	8	5.1	0.17	0	1.6	2.6	3.6	9

The columns are: the number of visits $|N|$, the number of staff members $|K|$, the number of synchronized visits $|P^{sync}|$, the average duration per staff member $\sum_i D_i/|K|$, average time to depot AvTD and the last five columns the average time window size for each group. The time windows for group A are actually shorter than 9 hours since each staff member has to reach the depot before the end of the 9 hours period. Looking at the average duration per staff, we note that the instances 4, 8 and 9 allow for much less waiting time than the other instances.

We use four different objective functions for the numerical experiments: minimize preferences, minimize traveling time, minimize maximal workload difference and for the larger instances, minimize the sum of traveling time and maximal workload difference. In the workload, we exclude traveling time and consider only the sum of durations.

4.2. Heuristic solution approach

The heuristic algorithm presented here is based on the idea of solving significantly restricted MIP problems to iteratively improve the best known feasible solution. In short, the restricted problems are supposed to be small enough (in number of variables and constraints) to result in a small B& B tree, and large enough to include an improved solution. We introduce dummy variables $d_i = 1$ if customer with visit i is not serviced by any vehicle and zero otherwise and adjust the constraints (1) accordingly. With the dummy variables penalized in the objective, the MIP has at least one trivial integer feasible solution.

The algorithm can be summarized in the following steps:

(1) Associate each visit with one or more vehicles and denote the current associations with $Y = \{(i,k)\text{: } k \text{ is allowed to visit } i\}$. The number of associations affects the solution time for the LP-relaxation in step 2.
(2) Solve the LP-relaxation of the model with the restriction to only allow associations in Y and let $\overline{A}$ be the subset of the arcs in A with a positive flow. Remove from Y the associations not utilized in the LP-solution.
(3) Solve the restricted MIP over Y and $\overline{A}$ until at least one feasible solution (utilizing dummy variables) is found.
(4) Repeat the following steps until an overall time limit is exceeded.
(5) Every R iterations reduce Y and $\overline{A}$ without removing arcs utilized by the best feasible solution found so far.
(6) Randomly extend Y with a selection of new associations and $\overline{A}$ with a selection of arcs.
(7) Solve the restricted MIP over Y and $\overline{A}$ with a time limit set to fit the current problem size.

We have chosen the set of parameters for the heuristic with the goal to find solutions for the numerical experiments and not to fine-tune the heuristic method. Many other settings provide similar performance. In our application of the algorithm, we set the overall time limit to 2 minutes for the small instances and 10 minutes for the larger

Fig. 4. The solution process for instance 8 over a time of 600 seconds and when we minimize the sum of traveling time and workload difference. The decreasing objective function value over time is shown with the unbroken line and is scaled on the left hand side y-axis with the unit hours. The increasing number of variables in the subproblems is printed with one dot for each iteration and the number of variables can be read on the right hand side y-axis.

instances. In the beginning when there are still dummy variables in use, we reduce the problem every second iteration (letting $R = 2$) using the probability 0.99 for an association to be removed from Y when the association does not involve customers serviced by dummy variables, and the probability 0.2 when it involves customers serviced by dummy variables. When a feasible solution, with $d_i = 0$, is found, we adjust the probability for associations to be removed to 0.8 every second iteration. Arcs not utilized in the current solution are removed in step 5 with a probability 0.5 and introduced in step 6 with a probability 0.05. The probability for associations to be introduced increases in every iteration with a factor 1.01 beginning with the probability 0.1. The time limit for the B&B solver in every iteration was set to $2|Y|/|K|$.

An illustration of the solution process for instance 8, when we minimize the sum of traveling time and workload difference, is shown in Fig. 4. The first solution not utilizing dummy variables was found after 111 seconds with the objective function value 28.0 hours. The dots in the figure mark the number of variables (read on the right hand side axis) and illustrate that our choice of probabilities gives an increased problem size over time.

4.3. Experiment settings

In every situation when we refer to CPLEX Branch and Bound, we refer to the following modification of the CPLEX default branching scheme. The model was extended with the help variables $y_{ik} = 1$ if vehicle k is serving customer with visit i and zero otherwise. In the branching scheme we prioritize branching on the help variables and thereafter branching on the routing variables using CPLEX default values.

The heuristic is comparable to the local branching heuristic (RDT), presented in Fischetti et al. (2004), in the following sense. In the model with the extension of help and dummy variables, we use the help variables as first level variables and the routing variables as second level variables. To fix a majority of first level variables to the values of a local solution, we select a random set of free

Table 4
Solutions proved optimal are marked in bold

Instance	TW	Preferences		Traveling time		Fairness	
		BK (hour)	H 2 min (hour)	BK (hour)	H 2 min (hour)	BK (hour)	H 2 min (hour)
1	F	**−96.45**	**−96.45**	**5.13**	**5.13**	**0.117**	**0.117**
1	S	**−114.03**	**−114.03**	**3.55**	**3.55**	0.026	0.052
1	M	**−117.80**	**−117.80**	3.55	3.55	0.026	0.026
1	L	**−118.51**	**−118.51**	3.44	3.39	0.026	0.026
1	A	−118.51	−116.37	3.16	3.69	**0.000**	0.026
2	F	**−85.26**	**−85.26**	**4.98**	**4.98**	**0.037**	**0.037**
2	S	**−92.09**	**−92.09**	**4.27**	**4.27**	0.025	0.025
2	M	−104.81	−102.63	3.58	3.58	0.025	0.025
2	L	−104.81	−106.06	3.58	3.42	0.012	0.025
2	A	−117.24	−117.24	3.58	3.34	0.012	0.012
3	F	**−56.70**	**−56.70**	**5.19**	**5.19**	**0.154**	**0.154**
3	S	**−99.49**	**−99.49**	**3.63**	**3.63**	0.064	0.064
3	M	**−106.59**	**−106.59**	3.41	3.33	0.038	0.064
3	L	**−107.87**	−104.72	3.29	3.29	0.038	0.013
3	A	−111.29	−92.22	3.1	3.28	0.038	0.026
4	F	**−63.08**	**−63.08**	**7.21**	**7.21**	**0.942**	**0.942**
4	S	**−100.00**	−99.43	**6.14**	6.69	0.130	0.162
4	M	−105.42	−105.42	5.91	5.75	0.130	0.049
4	L	−105.42	−96.96	5.83	5.3	0.081	0.032
4	A	−105.42	−92.78	5.23	4.91	0.032	0.065
5	F	**−62.59**	**−62.59**	**5.37**	**5.37**	**0.201**	**0.201**
5	S	**−76.29**	**−76.29**	**3.93**	**3.93**	0.063	0.038
5	M	**−76.29**	**−76.29**	**3.53**	**3.53**	0.038	0.063
5	L	**−84.21**	**−84.21**	3.43	3.34	0.025	0.025
5	A	−84.21	−43.74	3.26	3.45	0.025	0.038

first level variables. This hard fixing procedure represent the use of the refining constraint in RDT. We further simplify the subproblem by randomly excluding second level variables instead of using local-branching constraints as in RDT. In our heuristic, the diversification is performed by successively increasing the subproblems' size and the subproblem solver's time limit. The use of a random selection of free variables, instead of refining and local branching constraints, is because we prefer to reduce the number of binary variables active in each subproblem and solve the subproblems with a tighter time limit.

4.4. Solution quality from heuristic

We solved instances 1–5 with all variables with OPT and with a time limit of 60 minutes with the three objectives, minimize preferences, minimize traveling time and minimize the maximal difference in workload. To obtain as good benchmark solutions as possible within the time limit, we used the best found solution for a group of time windows as the start solution in the groups with larger time windows. The objective function values from these runs are shown in the BK columns, and the best known solutions (to be compared with), in Table 4. The heuristic algorithm (H 2 min) was always run without any initial solution. The solutions marked with bold face in the table were proved optimal within the time limit using OPT.

Out of the 75 instances solved, 33 were proved optimal. The heuristic found 29 of the optimal solutions and the heuristic found a better solution than the best known in 14 cases while it stopped with a worse solution in 18 cases. The increased number of multiple solutions introduced when we only use Fairness for the objective makes it more difficult to find bounds, which can be seen in the table where only the solutions with F sized time windows are proved optimal.

In Table 5, we compare the average of 20 runs on the first five instances with the solution found by OPT. The objective in these runs is to minimize the sum of traveling time and the maximum difference in workload over the instances with the group of large time windows (L). We exclude the larger instances 6–10 from this table because OPT found no feasible solution with all variables within the 60 minutes time limit.

None of the five solutions found with OPT proved optimal after 60 minutes. For instances 2 and 4, the average solution from the heuristic was better than the best OPT solution and for all instances the best solution from the heuristic was equally good or better than the OPT solution.

Table 5
The average, maximum and minimum objective function values from 20 runs of the heuristic (H 2 min) compared to the solution obtained from OPT after 2, 10, 30 and 60 minutes

	Instance				
	1	2	3	4	5
2 min	4.70	5.02	5.49	–	4.08
10 min	4.47	4.30	3.63	6.61	3.70
30 min	4.40	4.30	3.63	6.61	3.70
60 min	4.04	4.20	3.63	6.61	3.70
H 2 min, aver	4.29	4.09	3.94	6.41	3.83
H 2 min, max	4.88	4.39	4.38	7.77	4.33
H 2 min, min	4.01	3.96	3.63	5.88	3.70

4.5. Impact of synchronization constraints and time window size

In the following runs, we use the heuristic with an overall time limit of 10 minutes. The problems are solved to minimize the sum of traveling time and the maximum difference in workload with the large time window group L.

In Table 6, we show the results from 15 runs on instances 6–10, with the time for the synchronized visits fixed to the windows' midpoints in the column fix, and the synchronization constraints relaxed in the column no sync.

The column AvWT is the average workload and traveling time for all staff members in the solution as a percent of the 9 hours day. We found no feasible solution with the fixed time for the tightest problem instances (8 and 9). We experience from the results that with our model and with the heuristic presented in this paper, it is not more difficult to solve the problem with synchronization constraints than to

Table 6
Solutions when the synchronized visits are fixed to time windows' midpoints, synchronized with time windows, and where the synchronization constraints are relaxed

Instance	Fix		Sync		No Sync	
	Obj (hour)	AvWT (%)	Obj (hour)	AvWT (%)	Obj (hour)	AvWT (%)
6	11.97	64	11.87	64	10.59	62
7	14.16	71	11.52	68	12.97	69
8	–	–	15.16	84	13.78	83
9	–	–	20.68	81	19.29	80
10	17.69	68	17.61	68	16.35	67

Table 7
Solutions for the different time windows

Instance	S		M		L		A	
	Obj	AvWT	Obj	AvWT	Obj	AvWT	Obj	AvWT
6	13.69	66	12.80	65	11.87	64	11.88	64
7	15.06	72	13.45	70	11.52	68	12.41	69
8	–	–	–	–	15.16	84	13.01	82
9	–	–	–	–	20.68	81	22.89	81
10	16.24	67	15.33	67	17.61	68	17.59	67

solve the problem with relaxed the constraints (when we aim to find good feasible solutions).

In Table 7 we show the results with synchronization constraints for the S, M, L and A sized time windows with the larger instances 6–10.

We found no solution to the instances 8 and 9 with small and medium (S and M) sized time windows. We can observe two aspects of the new model. When we fix the time windows, it is difficult to find feasible solutions for the tight instances. By allowing larger time windows the solution quality improves. Although, with the time windows A, the heuristic would need a longer solution time to find better solutions than the settings where the visits are more time constrained. In our instances the average improvement using time windows L as compared to S, is substantial.

Table 8
Information and results about the additional test problems based on instances 1–5

$\|N\|$	$\|K\|$	$\|P^{sync}\|$	#var	#con	#inf
24	4	6	2600	2900	11
24	5	6	3300	3500	2
24	6	6	3900	4200	0
30	6	12	5900	6300	6
30	7	12	6900	7400	0
30	8	12	7800	8400	0
36	7	18	9700	10,300	12
36	8	18	11,000	11,300	1
36	9	18	12,400	13,200	1

The columns are the number of visits $|N|$, the number of staff members $|K|$, the number of synchronized visits $|P^{sync}|$, the number of variables #var the number of constraints #con and the total number of infeasible solutions (out of 25) for all time windows (F, S, M, L and A) #inf.

4.6. Impact of the proportion of synchronized visits

The instances 1–10 are based on the home care application and the proportion of synchronized visits is 10%. In order to study the behaviour of the model when this proportion increases we have constructed an additional set of test problems based on instances 1–5. In these new cases we study proportions 33%, 67% and 100%. We have simply added visits and required them to be synchronized with original visits. The number of staff members is adjusted so that each member has on average five customers to visit. In addition we solve with using one less and one additional member. Information on the test problems is given in Table 8.

Each row in the table is represented by all combinations of instances 1–5 and time windows, i.e. 25 test problems. In overall we have 225 test problems. The size of the problems (defined through the number of variables (column #var) and the number of constraints (column #con) varies with type of time window and in the table we give the dimension based on the largest problem. To solve each problem we use the heuristic with a limit of 10 minutes. The column #inf gives the number of test problems (out of 25) that fail to get a feasible solution. In total we have 33 such problems and they are represented through different time windows as follows: F-11, S-9, M-5, L-6 and A-3. By including one or two additional staff members we always find feasible solutions. This behaviour is also close to what is experienced from real planning situations in the home care case.

It is difficult to compare the effect of increasing the proportion of synchronization, since the number of synchronized visits give rise to different problems. To compare this effect, we assume that we have a general problem with m customer visits. We then, among this set of customer visits, increase the proportion of synchronized visits gradually up to 100%. In the case of 100% we still have the same m customer visits but all require synchronization (we only consider pairwise synchronization). The number of constraints will gradually increase as we add more constraints (7) (from the MIP model). The set of feasible solutions will decrease and the objective function value from the LP relaxation (and the lower bound) will increase (or be the same).

Moreover, with more constraints we get more structure that can be used in solution approach to reduce the problem size. These two aspects typically make the MIP problem easier to solve, which is observed earlier in our experiments. From the result we can observe that the number of infeasible solutions does not increase with the proportion of synchronized visits. We may therefore argue that the additional constraints do not make it more difficult to find feasible solutions. This is generally dependent on the application, but for the home care application there are many solutions available also when the synchronization proportion increases.

5. Concluding remarks

The proposed model is a generalization of the combined vehicle routing and scheduling model where temporal precedence and synchronization constraints are included. From the tests we have concluded that including synchronization constraints explicitly in the model has a positive effect on the planning. In the homecare application studied, more staff is introduced when no feasible solution is found among the regular staff. The size of the regular staff is dependent on the number of visits and the resources are tightly connected with the capacity required. Allowing a wide and flexible time window for synchronized visits may result in the fact that a feasible solution is found. Hence, no additional staff would be needed. The optimization based heuristic developed is efficient and finds high quality solutions within short time limits. This is true for instances with as well as without the additional constraints. Given a set of visits we have observed and argued that an increase of the proportion of synchronized visits does not make the VRP problem more difficult to solve.

The heuristic can also be used to schedule the synchronization visits before a decision support system is used (when the synchronization constraints are removed). The proposed model grows quickly (as standard VRP models) with the size of, i.e. number of visits, synchronized visits and staff members. For large instances of specific applications it is possible to develop other more efficient models, e.g. set partitioning/set covering models. One interesting example of future work is to use the proposed model as a basis for such model development in a similar fashion as in Ioachim et al. (1999). A second example is to generalize and develop the proposed heuristic approach for problems with similar structures.

References

Beck, J., Prosser, P., Selensky, E., 2003. Vehicle routing and job shop scheduling: what's the difference? In: Proceedings of the Thirteenth International Conference on Automated Planning and Scheduling (ICAPS03).

Bélanger, N., Desaulniers, G., Soumis, F., Desrosiers, J., 2006. Periodic airline fleet assignment with time windows, spacing constraints, and time dependent revenues. European Journal of Operational Research 175, 1754–1766.

Desrosiers, J., Dumas, Y., Solomon, M.M., Soumis, F., 1995. Network Routing, Handbook in Operations Research and Management Science. Time Constrained Routing and Scheduling, vol. 8. North-Holland, pp. 35–139, Chapter.

Eveborn, P., Flisberg, P., Rönnqvist, M., 2006. Laps care – and operational system for staff planning of home care. European Journal of Operational Research 171 (3), 962–976.

Fagerholt, K., Christiansen, M., 2000. A travelling salesman problem with allocation, time window and precedence constraints – an application to ship scheduling. International Transactions in Operational Research 7 (3), 231–244.

Fischetti, M., Polo, C., Scantamburlo, M., 2004. A local branching heuristic for mixed-integer programs with 2-level variables, with an application to a telecommunication network design problem. Networks 44 (2), 61–72.

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.

Gélinas, S., Soumis, F., 2005. Column Generation. In: Dantzig-Wolfe decomposition for job shop scheduling. Springer Science, New York, pp. 271–302, Chapter.

Haase, K., Desaulniers, G., Desrosiers, J., 2001. Simultaneous vehicle and crew scheduling in urban mass transit systems. Transportation Science 35 (3), 286–303.

Ioachim, I., Desrosiers, J., Soumis, F., Bélanger, N., 1999. Fleet assignment and routing with schedule synchronization constraints. European Journal of Operational Research 119, 75–90.

Karlsson, J., Rönnqvist, M., Bergström, J., 2004. An optimization model for annual harvest planning. Canadian Journal of Forest Research 34 (8), 1747–1754.

Palmgren, M., Rönnqvist, M., Värbrand, P., 2003. A solution approach for log truck scheduling based on composite pricing and branch and bound. International Transactions in Operational Research 10 (5), 433–448.

Sigurd, M., Pisinger, D., Sig, M., 2004. Scheduling transportation of live animals to avoid the spread of diseases. Transportation Science 38 (2), 197–209.