# Combined vehicle routing and scheduling with temporal precedence and synchronization constraints

David Bredström
Division of Optimization
Linköping Institute of Technology
Email: dabre@mai.liu.se

Mikael Rönnqvist
Department of Finance and Management Science
Norwegian School of Economics and Business Administration
Email: mikael.ronnqvist@nhh.no

**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 the temporal constraints, in the literature, usually are remarkably simplified in the solution process, even though these constraints may significantly improve the solution quality and/or usability. 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 usage of a commercial solver against the proposed heuristic where the latter approach can find high quality solutions within distinct 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 is frequently occurring in well known problems such as the dial-a-ride and the pickup and delivery problems. However, the problem with vehicle dependencies is given much less attention in the literature although its wide range of practical applications. 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, a set of customers to be served within their respective prescribed time window, the objective for the vehicle routing and scheduling problem (VRSP-TW) is for example to minimize the total traveling time. Both heuristic and exact solution methods have been suggested to solve 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 the live animal transport.

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 normally to execute 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 the study they include vehicle independent precedence constraints to the VRP and observe that the routing techniques they use suffer from difficulties to find feasible solutions, while the scheduling techniques find feasible solutions to all the studied problem instances.

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 to a relief points 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 caretakers. The model for periodic routing and airline fleet assignment problem presented in the paper of 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 same sense as 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 contribution 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 strict limit the time windows providing a simplified VRP problem. A model that consider some synchronization constraints for an airline fleet assignment and routing is given in Ioachim et al. (1999). Our model is however more general and based on an extension of a traditional VRP model. We also demonstrate through the computation experiments that the proposed model is not significantly harder to solve as compared to the simplified application using a VRSP-TW model. 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 distinct time limits. We do not suggest that this model should be used directly to solve all applications. It does however describe the temporal constraints in a clear way. It can also be used as a basis when formulating and developing more application oriented models and solution methods.

This paper is outlined as follows. In Section 2 we describe the problem and illustrate with an illustrative example. Here is also a description of some typical applications where precedence and synchronization constraints are an important aspect. In Section 3 we provide the new model. We focus on constraints that relates 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. It is easy to include but do not add any to the model. In Section 4 we describe the numerical tests done. This includes a description of the test problems, a description of the developed heuristic and analyses of the tests. Finally, we give some concluding remarks and outline some further work in Section 5.

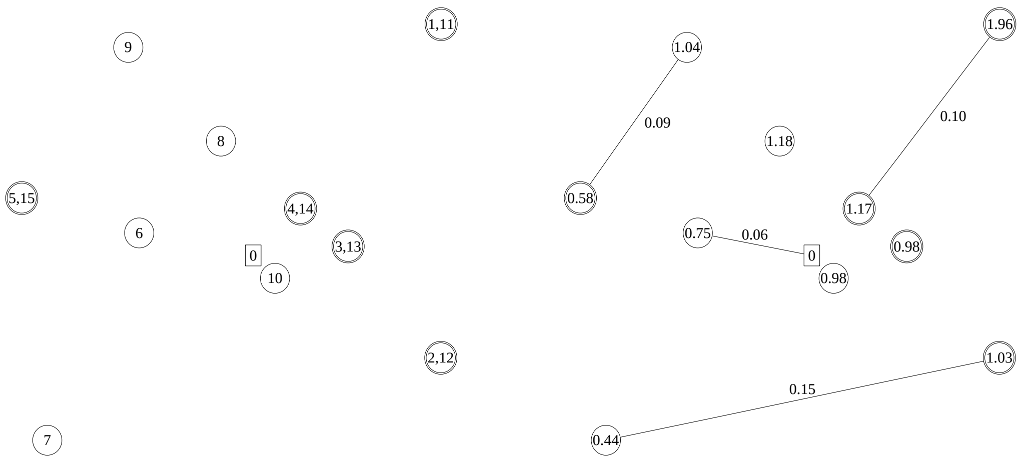
## 2 Problem formulation

We assume to have a fleet of vehicles available in a depot, and a set of customers to be visited and served within their respective prescribed time window. Let $K$ denote a set of vehicles and let $G = (\bar{N}, A)$ be a directed graph, where $\bar{N} = \{o, d, 1, \ldots, n\}$ is the node set and $A = \{(i,j) \,|\, i \neq j, i \in \bar{N} \setminus \{d\}, j \in \bar{N} \setminus \{o\}\}$ is the arc set. The nodes $o$ and $d$ both represent the depot and the nodes $N = \{1, \ldots, n\}$ is the customers to be visited. Each customer $i \in N$ 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 for 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 $P^{sync} \subset N \times N$, and the nodes with pairwise precedence constraints with $P^{prec} \subset N \times N$. For each pair $(i,j) \in P^{prec}$ we define a temporal offset $S_{ij}$ if it is required that $j$ is visited at least $S_{ij}$ time units after $i$. We call a customer $j$ virtual in a pair $(i,j) \in P^{sync}$, or $(i,j) \in P^{prec}$, when $i$ and $j$ refer to one customer.

## Example

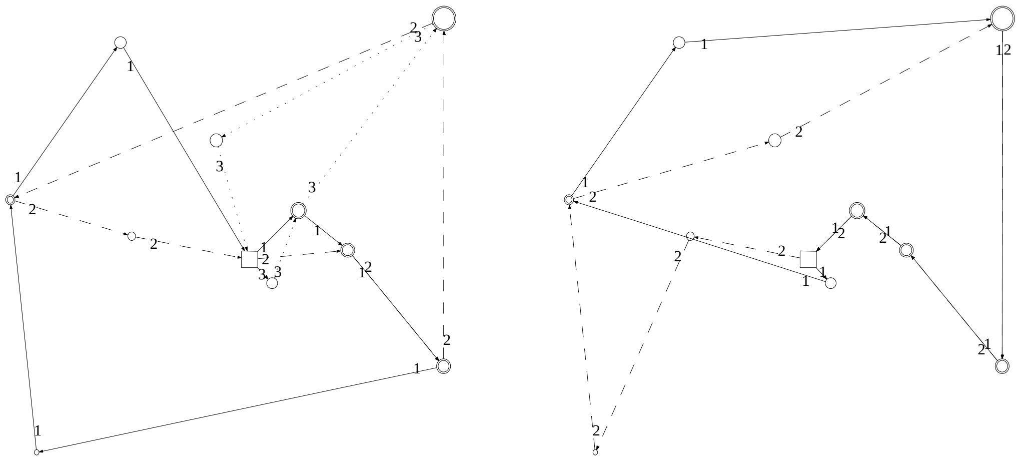
In the example we use a network with ten physical customers and where five customers needs service from two simultaneous vehicles. We have $N = \{1, \dots, 15\}$ where $11, \dots, 15$ are virtual customers in the pairs $(1,11), \dots, (5,15) \in P^{sync}$. In the Figure 1(a) we show the network with ten nodes where the synchronized visits are indicated with a double circle. The service durations are shown in the Figure 1(b) 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.




(a) The network. (b) Service durations.

Figure 1: Example network and durations.

We illustrate the optimal solutions for two problems solved on the network in the Figure 2 where in both problems the vehicles have an availability of nine hours and the customer time windows are disregarded. In the first example, as seen in Figure 2(a) with the schedules in Table 1, we use three vehicles with the objective to minimize the sum of the traveling distance and the maximal difference in workload, 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 Figure 2(b) 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 hours. The maximal differences in total duration for the vehicles is only 0.05 hours in the first example.



(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.

Figure 2: Example solutions to two problems on one network.

| Vehicle 1 | Arrival | Duration | Traveling | Waiting |
|---|---|---|---|---|
| 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 | Arrival | Duration | Traveling | Waiting |
| 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 | Arrival | Duration | Traveling | Waiting |
| 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 |

Table 1: Schedules for the three vehicles in the first example.

## 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

| Vehicle 1 | Arrival | Duration | Traveling | Waiting |
|---|---|---|---|---|
| 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 | Arrival | Duration | Traveling | Waiting |
| 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 |

Table 2: Schedules for the two vehicles in the second example.

time such 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 constraints are handled. We start by describing a homecare staff scheduling problem and then move to two airline planning problems and two forest operations problems. We describe the homecare problem in more detail because it is the basis for the numerical experiments.

## Homecare staff scheduling

Daily planning of homecare staff is a combined scheduling and routing problem. The planning problem is to establish routes for staff member 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 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 begins and ends the day in one base position where the final alterations to plans are made and 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 a elderly person living at home but has an agreement with the local community about support for e.g. cleaning, bathing, cooking or medical attention. If the visit is for cooking and say one hour the time window may be limited to that the visit starts between 10.30 am and 11.30 am. If the visit is for cleaning the start may be anywhere between 8.00 am and 3.00 pm. 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 its own route. However, in some customer visits there is an requirement that two staff members occurs simultaneously or in given order. This is due to, for example, 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 percent of the tasks follows such requirements. This create a big 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 split the visit into two, the original and a virtual. Each must be done by a staff member and in order to make it at the same time, the time window for the start of the visit is set such that there is only a 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 within 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 planning of security guards. Visits in this context is to visit a location, e.g., an industrial site and check doors, windows and locks. Each guard has a skill depending on e.g. if she or he has a dog or trained for specific tasks at customers. The synchronization constraints arises when there is a need for more than one guard to make a visit. Precedence constraints arise when there is a need to make say three different and ordered visits to the same location but there is a freedom when they are done.

### Airline scheduling

There are many scheduling problems arising within the airline industry and there are numerous articles published. We will describe two applications where synchronization constraints are important aspects. The first is a fleet assignment problem and the second a crew scheduling problem. Given a fixed flight timetable, there is a need to decide which aircraft type to use on each flight on a daily basis. Each aircraft type available has a given capacity and the objective is to match the capacity with the expected demand of passengers on all flight. This problem is described in e.g. Hane et al. (1997). However, if the timetable is not entirely fixed and there is a flexibility in changing departure times (within time windows) the problems becomes more complex. This problem is studied in Ioachim et al. (1999). They study the case to assign airplanes to flights on a daily basis but for a time period of an entire week. The synchronization constraint states that the same flight (but in different days) needs to departure on the same time. The reason for this is to get a more robust schedule. The alternative approach is to solve the problem for each individual day. The used objective would be better (or same) but different departure times would be confusing and be less useful. The solution approach is based on formulating the problem including the synchronization constraints as a multi-commodity flow formulation and a solution method based on a side constrained set partitioning reformulation. This is then solved with column generation in a branch-and-bound framework. This is, as mentioned earlier, one of the few where synchronization is included. The solution process is further developed in Bélanger et al. (2006).

Given the fleet assignment it is known which aircraft type that will be used on each flight. Each aircraft type has given description and requirements on cockpit and cabin crew. The crew pairing problem is to decide routes for generic crews. This is later used to decide the crew assignment i.e. decide schedules for individual crew members. The traditional approach in crew pairing is to decide the routes for each type of aircrafts. For example, Boeing 747 (B747) aircrafts are planned separately from Boeing 767 (B767) aircrafts. Cockpit crews they need to be planned separate because of particular skills and training. However, cabin crew over several types could be planned together since the same skills are needed. The problem of coordinated planning is the difference in crew size. Suppose we have crews of either 16 members (B747) or 9 members (B767) members. Then we could split the larger B747 crews into two subgroups, 9+7 members. In this way the subgroup of 9 members from a B747 could be scheduled to work also on a B767. Integrated planning will provide increased possibilities for better pairing. The difference with traditional planning is that there is a synchronization constraint stating that two B747 subgroups (one with 9 and one with 7 members) must be assigned to each B747 flight. It is also possible to assign a full 16 member B747 crew.

### 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 include e.g. description of stands and demands at mills. A description of the problem, models and methods is provided in Karlsson et al. (2004). A 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 need a particular time to perform the operations at each stand. Time and cost to move a vehicle depends on the distance between stands. If the distance is short, typically less than 5 km, the vehicles drives themselves. 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 the capacity can be utilized efficient. A precedence restriction is that forwarding can be done once the harvesting is done. Also, 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 combine harvesters and forwarders to 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 time for a stand the performance is limited to the slowest, for each stand.

Routing of logging trucks are 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 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 to not have a crane is an increased weight capacity, typically 5-10%. Loaders are only located to areas where there is a certain level of harvesting. These loaders often serve several stands and are moving between these in order to load trucks. To find an optimal plan the routes of the trucks needs to be planned together with the routes of the loaders. A synchronization constraint is that a loader must be at a stand when the truck arrives. In practice, the loaders are planned separately. Given the schedule of the loaders, a set of time windows when the stands 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 $t_{ik} \geq 0$. The routing variable $x_{ijk}$ is one if the vehicle $k \in K$ is traversing the arc $(i,j) \in A$. The scheduling variable $t_{ik}$ is the time the vehicle $k$ arrives to the customer $i \in N$ and is zero if the vehicle $k$ does not visit the customer $i$. The routing and scheduling constraints are modeled as follows.

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

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

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

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

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

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

The constraints (1)-(5) form the constraint set for a multiple traveling salesman problem, where, if we use the vocabulary of the VRSP, the constraints (1) ensures that each customer is visited by exactly one vehicle, (2) and (3) define the routing network, and the constraints (4) - (6) are the scheduling constraints. The constraint (5) implies that $t_{ik} = 0$ if customer $i$

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.

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

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

The constraints (7) ensures that the vehicles that visit the customers $i$ and $j$ for $(i,j) \in P^{sync}$ arrive simultaneously. The customer $j$ in a pair $(i,j)$ is typically a virtual customer representing the need for simultaneous service from a second vehicle to the customer $i$. Universally, we can model the demand of $s$ vehicles for one customer by introducing $s-1$ virtual customers $i_2,\dots,i_s$ and the relations $(i_1,i_2)$, $(i_1,i_3),\dots,(i_1,i_s) \in P^{sync}$.

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 to one customer $j$ before the service of another specific customer $i$ is finished. This requirement is modeled by letting $S_{ij} = -D_i$. Another situation is when we have a demand of a second vehicle to arrive while a customer is served. For example, in the homecare application this is the situation when one staff member is visiting a care taker 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 $(i,j)$, $(j,i) \in P^{prec}$. It is worth noting that a synchronization constraints 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 decompose in one problem for each vehicle and this fact is used in many heuristic and exact solution methods. The 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 we in many applications prefer to avoid the vehicle independent temporal constraints.

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

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

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 we have the objective 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

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

$$s.t. \quad (1) - (9) \tag{11}$$

where $\alpha_P, \alpha_T$ and $\alpha_B$ are non-negative weights and $c_{ik}$ is a preference measure for vehicle $k$ to serve customer $i$.

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

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

We can also consider an objective to minimize waiting time, if we for instance 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 with the numerical tests is to analyze the behavior of the model and the usefulness of including the new constraints explicit 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 using a maximum of 2 GB RAM. When we solved some of the larger instances we found that CPLEX had problems to find solutions in reasonable time. It is well known that the MIP formulation used in this paper of a VRSP problem has a large MIP-GAP and this is generally also the case when we have additional temporal constraints. We therefore introduce an optimization based heuristic approach that we use in the tests. This heuristic is based on 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 the heuristic. Then we test the performance of the heuristic on the smaller instances. We also make some analyses of the model characteristics. Then we turn to solve some larger instances and the impact of synchronization constraints and time window size.

### Test problems

We assumed that there are on average five customers for each staff member to visit, of which in total 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 nine hours, and we excluded the scheduling of rests. We generated five small sized instances used for benchmarking the heuristic algorithm and five realistic sized instances as presented in the 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.

| Instance | $\|N\|$ | $\|K\|$ | $\|P^{sync}\|$ | $\sum D_i/\|K\|$ (h) | AvTD (h) | F (h) | S (h) | M (h) | L (h) | A (h) |
|---|---|---|---|---|---|---|---|---|---|---|
| 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* |

Table 3: Test instances. 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 hour 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 note that the average duration time is largest in instance 4, 8 and 9. This implies that these are most tight and generally more difficult to solve. The instances 1-5 are of order $1,900$ variables and $2,100$ constraints, the instances 6-8 of order $27,000$ variables and $28,000$ constraints, and the largest instances 9-10 of order $106,000$ 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 to have a mean of five hours workload (excluding traveling time) for each staff member. The traveling time and durations were rounded to integers and measured in minutes. The network for instance 8 is illustrated in Figure 3.

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

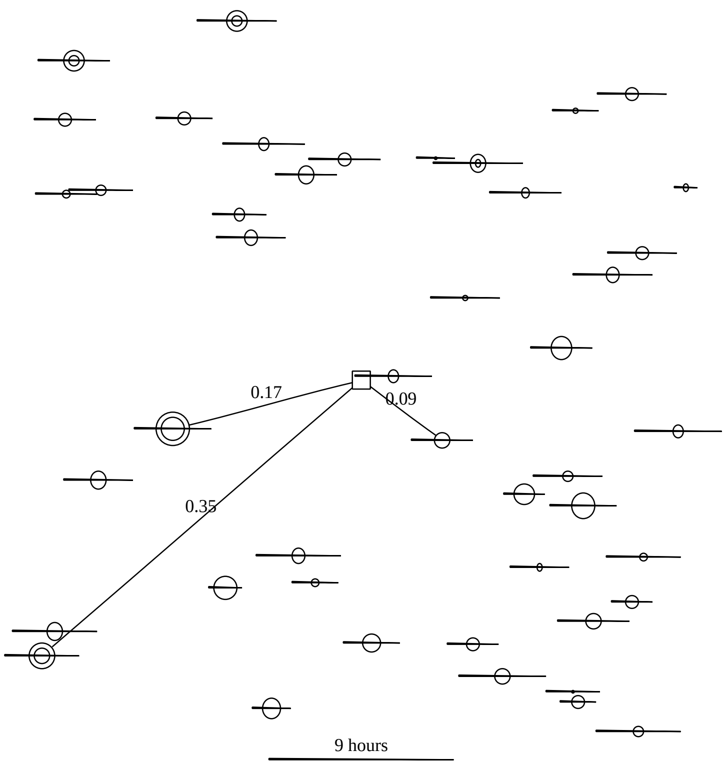



Figure 3: Instance 8 with time windows (for arrival time) from group L shown proportionally to the durations by the length of the crossing bars. The circle diameter is relative to the visit's duration and the scale relative to nine hours, printed for reference with the bottom line. The double circles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

## Heuristic solution approach

The heuristic algorithm presented here is based on the idea to solve 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 the customer $i$ is not served 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 customer with one or more vehicles and denote the current associations with $Y = \{(i,k) : k \text{ is allowed to visit } i\}$. 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 $\bar{A}$ 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 $\bar{A}$ until at least a feasible solution (utilizing dummy variables) is found.

4. Repeat the following steps until an overall time limit is exceeded:

5. Every $R$s iteration reduce $Y$ and $\bar{A}$ without excluding arcs utilized by the best feasible solution found so far.

6. Randomly extend $Y$ with a selection of new associations and $\bar{A}$ with a selection of arcs.

7. Solve the restricted MIP over $Y$ and $\bar{A}$ 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. In our application of the algorithm we set the overall time limit to two minutes for the small instances and ten minutes for the larger instances. In the beginning when there still are 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 served by dummy variables, and the probability 0.2 when involving customers served 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 is increasing in every iteration with a factor 1.01 beginning with the probability 0.1. The time limit for the B&B solver was in every iteration set to $2|Y|/|K|$.

An illustration of the solution process for the instance 8 with the objective to minimize the sum of traveling time and workload difference is shown in Figure 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 increase the problem size over time.

### Experiment settings

In all situations when we refer to the Branch and Bound tree, 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 the customer $i$ and zero otherwise. In the branching scheme we prior branching on the help variables and thereafter branching on the routing variables using CPLEX default values.

The heuristic is based on the local branching heuristic (RDT), presented in Fischetti et al. (2004). In the model presented in this paper we have the help variables as first level

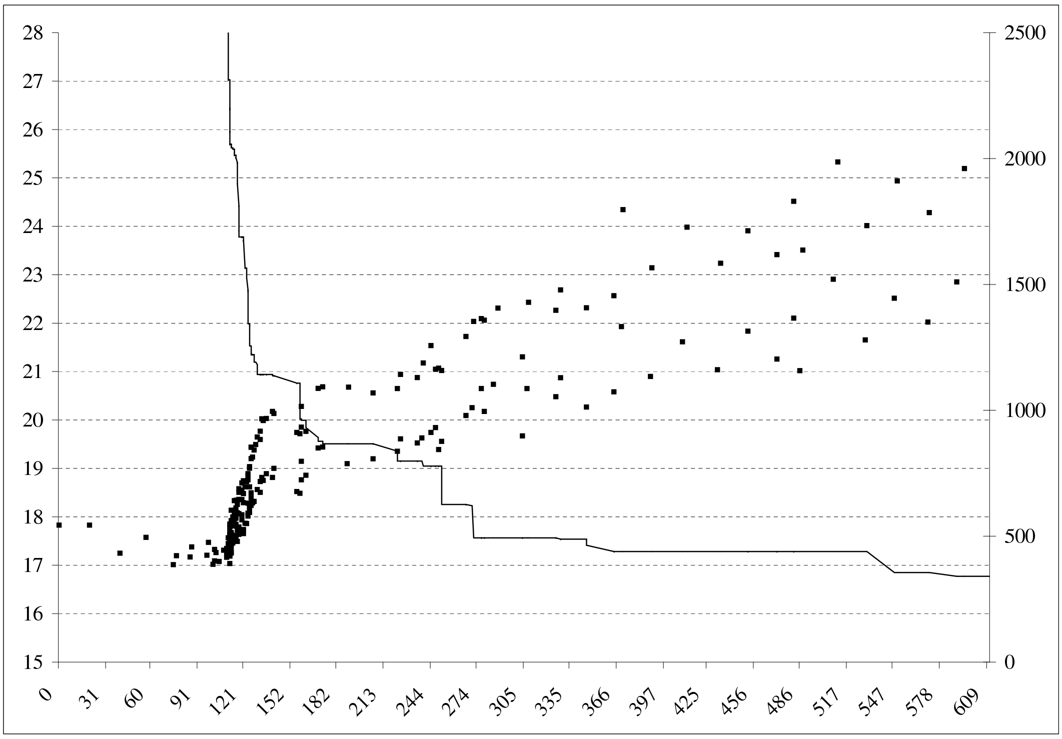



Figure 4: The solution process for the instance 8 during 600 seconds with the objective to minimize the sum of traveling time and workload difference. The decreasing objective function value over time is shown with the unbroken line and scaled on the left hand side $y$-axis with the unit hours. The increasing 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.

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 first level variables, compared to 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 their work. The diversification is in our heuristic performed by successively increasing the subproblems size and subproblem solver's time limit. The use of 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.

## Solution quality from heuristic

We solved the 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 start solution in the groups with larger time windows. The objective function values from these runs are shown in the BK columns, the best known solutions (to be compared with), in the 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

| | | Preferences | | Traveling time | | Fairness | |
|---|---|---|---|---|---|---|---|
| Instance | TW | BK (h) | H 2 min (h) | BK (h) | H 2 min (h) | BK (h) | H 2 min (h) |
| 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 |

Table 4: Solutions proved optimal are marked with a bold face.

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 was 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 the Table 5 we compare the average of 20 runs on the first five instances with the solution found by OPT. The objective is in these runs 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 where proved optimal after 60 minutes. For the 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.

| 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 |

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

### 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 with the objective to minimize the sum of traveling time and the maximum difference in workload and the large time window group L.

In Table 6 we show the results from 15 runs on the 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 the average workload and traveling time for all staff members in the solution in percent of the 9 hour day. We found no feasible solution with fixed time for the most tight problem instances 8 and 9. With the model and using the heuristic presented in this paper it is not more difficult to solve the problem with synchronization constraints compared to relaxing the constraints.

| Instance | Fix | | Sync | | No Sync | |
|---|---|---|---|---|---|---|
| | obj (h) | AvWT (%) | obj (h) | AvWT (%) | obj (h) | 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 6: Solutions when the synchronized visits are fixed to time windows' midpoints, synchronized with time windows, and where the synchronization constraints are relaxed.

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 even find feasible solutions for the tight instances. These would be most close to the real situation. By allowing larger time windows the solution quality improves. Although, with the time windows A, the heuristic would need 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.

| Instance | S | | M | | L | | A | |
|---|---|---|---|---|---|---|---|---|
| | obj (h) | AvWT (%) | obj (h) | AvTW (%) | obj (h) | AvTW (%) | obj | AvTW (%) |
| 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 |

Table 7: Solutions for the different time windows.

# 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. In the tests we have shown that including synchronization constraints explicit 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 regular staff size is dependent on the number of visits and the resources are tightly connected with the required capacity. Allowing a wide and flexible time window for synchronized visits may result in 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. 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 quick (as standard VRP models) with the size 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 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 is to further test and develop the proposed heuristic approach either by itself or in a combination with a more traditional Branch & Bound algorithm.

# References

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

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

J. Desrosiers, Y. Dumas, M. M. Solomon, and F. Soumis. *Network Routing*, volume 8 of *Handbook in Operations Research and Management Science*, chapter Time Constrained Routing and Scheduling, pages 35–139. North-Holland, 1995.

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

K. Fagerholt and M. Christiansen. 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, 2000.

M. Fischetti, C. Polo, and M. Scantamburlo. 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, 2004.

R. Freling, D. Huisman, and A. P. M. Wagelmans. Models and algorithms for integration of vehicle and crew scheduling. *Journal of Scheduling*, 6:63–85, 2003.

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

C. Hane, C. Barnhart, E. Johnson, R. Marsten, G. Nemhauser, and G. Sigismondi. The fleet assignment problem: Solving a large-scale integer program. *Mathematical Programming*, 70:211–232, 1997.

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

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

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

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