A hybrid method based on linear programming and tabu search for routing of logging trucks

Patrik Flisberg[†], Bertil Lidén[†], Mikael Rönnqvist[†*]

[†]The Forestry Research Institute of Sweden,
SE-75183 Uppsala, Sweden

[*]Norwegian School of Economics and Business Administration,
N-5045 Bergen, Norway

Abstract

In this paper, we consider an operational routing problem to decide the daily routes of logging trucks in forestry. The industrial problem is difficult and includes aspects such as pickup and delivery with split pickups, multiple products, time windows, several time periods, multiple depots, driver changes and a heterogeneous truck fleet. In addition, the problem size is large and the solution time limited. We describe a two-phase solution approach which transforms the problem into a vehicle routing problem with time windows. In the first phase, we solve an LP problem in order to find a destination of flow from supply points to demand points. Based on this solution, we create transport nodes each of which defines the origin(s) and destination for a full truckload. In phase two, we make use of a standard tabu search method to combine these transport nodes, which can be considered to be customers in vehicle routing problems, into actual routes. The standard tabu search method is extended to consider some new features. The solution approach is implemented as part of a newly developed decision support system and we report on tests made on a set of industrial cases from major forest companies in Sweden.

keywords: Forestry, Routing, Tabu search, Linear Programming, OR in Practice

1 Introduction

The routing of logging trucks is an increasingly important problem for forest companies. This is based on both economic gain and on increasing environmental concern. Many forest companies are aware of the increased efficiency which can be obtained through improved supply chain management. This often includes integrated and better planning of their own or their subcontractors' fleets of trucks. The planning of logging trucks has traditionally been a manual process performed by the transport planners responsible for a small number of trucks over a specified and limited region. Better planning is achieved using larger regions and a larger truck fleet. However, this larger and more difficult planning requires some decision support system (DSS) which can handle all the information and provide solutions to the routing problem.

The actual routing problem can be described as follows. There is a supply of different assortments at harvest areas in forests and at industries a demand for assortment groups, each of which may include several assortments. The volumes at a supply may vary from a fraction of a truckload to many truckloads. There are time windows at both the industries and the harvest areas. Demand at an industry is typically given on a weekly basis, whereas routes are to be found on a daily basis. The routing is a pickup and delivery problem where it may be necessary to pickup several small piles of supplies in order to get a full truckload. Each truck has a given home base and working hours. Most trucks change driver at least once during the day at a change-over location. A truck may be equipped with its own crane or require a loader to be present for loading and unloading.

Beside this being a very difficult planning problem there are additional complicating factors involved in the development of a DSS for routing logging trucks. There is a need for detailed information about roads, for example, distances, speed limits and road quality. A second factor has been the need to have access to accurate information about truck availability, demand and in particular, supply. These two factors have been included in a system called RuttOpt which is a DSS developed by the Forestry Research Institute of Sweden (Skogforsk). A third factor has been the need to have quick and robust methods that can assist the planner with detailed routes that are cost effective for the entire fleet. Solution methods have been under continual development but so far they have not been able to cope with the general problem studied in this article.

Many methods have been proposed for the vehicle routing problem (VRP). However, due to the fact that it is a hard combinatorial problem, exact methods perform poorly for real size problems and this motivates the development of metaheuristics. Moreover, there are many versions on VRP which take into account a variety of aspects such as pickup and delivery, backhauling, multiple depots, heterogeneous fleet, multiple routes per vehicle etc. For general surveys of VRP we refer to Cordeau et al. [4] and Gendreau et al. [10]. Bräysy and Gendrau [3] provide a survey of methods for VRP with time windows. Ropke and Pisinger [15] develop an adaptive large neighbourhood search heuristic for the pickup and delivery problem with time windows. Archetti et al. [1] use a tabu search method for a split delivery VRP and Ho and Haugland [13] study split deliveries with time windows. Ropke and Pisinger [16] provide an overview of VRP with backhauls and suggest a unified heuristic. This also covers pickup and delivery problems with time windows. Crevier et al. [6] study a multi-depot problem where the vehicles are allowed to make stops at intermediate depots in order to be replenished.

There are some specific aspects to the routing problem of logging trucks which makes it different from a standard VRP. Supply volumes are generally larger than actual demand. Furthermore, there is generally no specified linkage that states that a specific supply should be transported to a specific demand. Typically all supplies can be used to transport to all demands providing correctness of assortments. The demand typically ranges over several days but has lower and upper limits for each day. This implies an integration between days or time periods and the problem becomes a multi period problem. In our application, each truck has as many routes as there are days. Furthermore, companies (and countries) exhibit large differences in how the decision process takes place and what restrictions are included. These differences are based on a variety of factors including company management, organisation of trucks, information available and usage of geographical information systems (GIS).

An early DSS for logging trucks is ASICAM (Weintraub et al. [7]) which is used by several forest companies in Chile and other South American countries. It produces a schedule for one day by a simulation based heuristic that assigns transport orders (combination of pickup and delivery) to trucks in a moving time horizon. An example of a decentralized system is Åkarweb (Eriksson and Rönnqvist [8]). Åkarweb is a web based system that each day computes potential transport orders by solving a Linear Programming (LP) based backhauling problem. From this system, transport managers select transport orders to combine them into routes. In Gingras et al. [11] a system named MaxTour for forest routing in Quebec, Canada, is described. This system establishes routes based on the classical heuristic by Clarke and Wright by combining predefined loads in origin-destination pairs.

In Palmgren et al. [18] a column based routing model is used and solved using Branch & Price. The pricing process (column generation) is based on a pre-generated pool of columns. This pool is found by a heuristic enumeration which in turn uses the result from a LP based flow problem. In Palmgren et al. [19] the same approach is used but the pool is extended by resolving the LP problem several times. Murphy [14] formulates a general integer programming model for the routing model, but uses it only for tactical long term planning. Gronalt and Hirsch [12] describe a tabu search method where a set of fixed transports are to be performed. Time windows and multiple depots are included in the formulation. Dispatching involves deciding routes (or parts of routes) continuously during the day based on real time events such as queuing, bad weather, truck break down etc. In Rönnqvist and Ryan [21] a solution method for dispatching is described. The method establishes solutions for a fleet of trucks within a few seconds. It is based on recursively solving a column based model whenever changes in data occur. In Rönnqvist et al. [22], a similar dispatch problem is studied with a method based on recursively solving an assignment problems.

The actual methods implemented in DSS and used in practice at forest companies are typically quite simple. Often the systems support manual planning but full optimization is not embedded. In developing a solution methodology we need to consider the size of the problem and the limit demanded on the solution time. A standard but small case would have about 10-15 trucks (some with and some without a crane), 500 supply points, 20 demand points and a planning period of

1-5 days. The requirement or the desired solution time is to find a high quality solution within 15 minutes on a standard PC. However, we also need to solve case studies with up to 110 trucks, 2,500 supply points and 100 demand points and with a planning period of five days.

We propose a solution approach in two phases. In the first phase we construct so called *transport nodes*. A transport node describes the possible multiple pickup points and one delivery point for a full truckload. This is done by solving a flow problem using variables for each truck and each combination of supply and demand points. Constraints describe demand, supply and the times when each truck is available. We also use transport nodes to describe the change of drivers during a day. Given the transport nodes, we can formulate a VRP problem with time windows (VRPTW). In standard VRP terms, a transport node represent one customer. In the second stage, we use a well known tabu search method as a basis for combining transport nodes to routes. We make use of the unified tabu search method proposed in Cordeau et al. [5]. The method is extended to enable differences in supply and demand and multiple home bases. The usage of two phases is similar to the one described by Bent and Van Hentenryck [2] who develop a two-stage hybrid algorithm for pickup and delivery VRP with time windows. In their article, the first stage is used to limit the number of routes and in stage two, a large neighbourhood search is used to find the routes. The method proposed in this article is implemented in RuttOpt and we show results from two case studies from Swedish forest companies. Results include a case with 110 trucks, 113 demand points and 2531 supply points. For planning over five days, this represents more than 3,800 customers.

The outline of the paper is as follows. In section 2 we describe the routing problem for logging trucks in forestry. In section 3 we describe the proposed solution approach and the models used. In section 4, we show the results obtained from testing the model on real data from two Swedish forest companies. In section 5 we make some concluding remarks.

2 Problem description

The route planning of logging trucks is one part of the forest supply chain (Rönnqvist [20]). The routing consists of deciding a cost effective schedule, one route for each truck, to match demand with supply. The supply is described by the actual piles of logs stored adjacent to forest roads and the demand is a detailed description of industrial orders. It is an operational planning problem with a planning period of one day to one week. The reason for using one week is that industrial demand is often expressed on a weekly basis, but a certain proportion needs to be delivered during each day.

The operational planning is related to the tactical planning and should follow a tactical destination plan. The destination problem is generally solved using a monthly planning period in which the catchment areas for each combination of industry and assortment are determined. These can be found by solving a flow problem where the flow between harvest areas and industries is determined. The solution can be used for example to distribute work fairly between haulage companies. In Forsberg et al. [9], a system for a tactical problem is described. Here, train and ship transportation and backhaul planning are included.

The destination planning is often done centrally by a forest company which is responsible for deliveries to industries. Once a destination plan has been found, transport orders are distributed to a number of transporters. Transporters may be a combination of a larger independent transport company, a transport organisation within the forest company or individual hauliers with one or a few trucks. Each transporter typically operate within a specific area. The distribution of transport orders together with decentralized planning limit how good the routes generated can be. Interest in planning for larger truck fleets, which leads to large potential savings, is increasing.

In figure 1 we start to describe the actual routing to show a typical route performed by a logging truck during one day. Detailed information about a route is given in table 1. We note that it is a pickup and delivery problem. A truckload for a delivery to a customer may be picked up at several supply points. There may also be several pickups at the same supply point during the day. In the example, there is also a change of drivers.

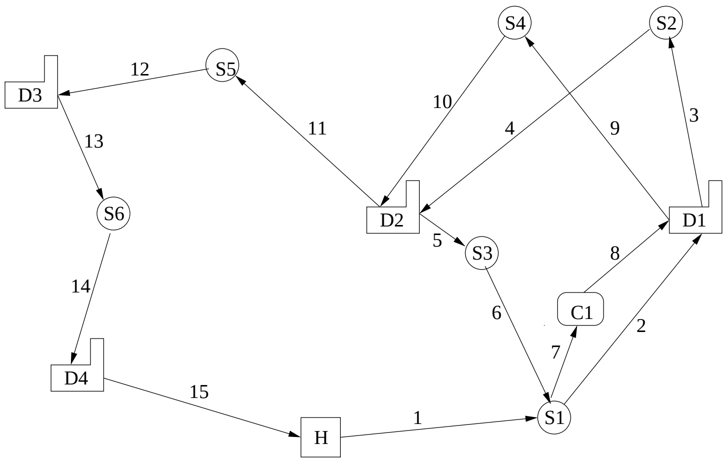

Figure 1: Example of a daily route for a logging truck. Driving between locations is numbered from 1 to 15.

Segment	Time	From	Time	To	Operation	Assortment	Volume
1	06.00	H	06.45	S1	Drive to harvest area S1	—	—
	06.45	S1	07.00	S1	Load logs at S1	Spruce sawlogs	40 ton
2	07.00	S1	07.50	D1	Drive to saw mill D1	Spruce sawlogs	40 ton
	07.50	D1	07.10	D1	Unload logs at D1	Spruce sawlogs	40 ton
3	07.10	D1	08.00	S2	Drive to harvest area S2	—	—
	08.00	S2	08.30	S2	Load logs at S2	Spruce pulplogs	40 ton
4	08.30	S2	09.50	D2	Drive to pulp mill D2	Spruce pulplogs	40 ton
	09.50	D2	10.00	D2	Unload logs at D2	Spruce pulplogs	40 ton
5	10.00	D2	11.00	S3	Drive to harvest area S3	—	—
	11.00	S3	11.20	S3	Load logs at S3	Spruce sawlogs	25 ton
6	11.20	S3	11.40	S1	Drive to harvest area S1	Spruce sawlogs	25 ton
	11.40	S1	12.00	S1	Load logs at S1	Spruce sawlogs	15 ton
7	12.00	S1	12.30	C1	Drive to change node C1	Spruce sawlogs	40 ton
	12.30	C1	12.40	C1	Change driver	Spruce sawlogs	40 ton
8	12.40	C1	13.30	D1	Drive to saw mill D1	Spruce sawlogs	40 ton
	13.30	D1	13.50	D1	Unload logs at D1	Spruce sawlogs	40 ton
9	13.50	D1	15.00	S4	Drive to harvest area S4	—	—
	15.00	S4	15.20	S4	Load logs at S4	Spruce pulplogs	20 ton
	15.20	S4	15.40	S4	Load logs at S4	Pine pulplogs	20 ton
10	15.40	S4	17.00	D2	Drive to pulp mill D2	Spruce/ Pine pulplogs	40 ton
	17.00	D2	17.20	D2	Unload logs at D2	Spruce/ Pine pulplogs	40 ton
11	17.20	D2	18.40	S5	Drive to harvest area S5	—	—
	18.40	S5	19.10	S5	Load logs at S5	Spruce sawlogs	40 ton
12	19.10	S5	20.00	D3	Drive to saw mill D3	Spruce sawlogs	40 ton
	20.00	D3	20.30	D3	Unload logs at D3	Spruce sawlogs	40 ton
13	20.30	D3	21.20	S6	Drive to harvest area S6	—	—
	21.20	S6	22.00	S6	Load logs at S6	Pine pulplogs	40 ton
14	22.00	S6	22.50	D4	Drive to pulp mill D4	Pine pulplogs	40 ton
	22.50	D4	23.10	D4	Unload logs at D4	Pine pulplogs	40 ton
15	23.10	D4	23.40	H	Drive home	—	—

Table 1: Information of a typical route during one day related to the route in figure 1.

2.1 Planning components

Supply and assortments

At each harvest area a number of products or assortments are produced. An *assortment* is defined by species, for example Spruce, Birch or Pine, together with dimensions and quality. Logs with a smaller diameter are typically pulplogs and logs with a larger diameter are classified into different sawlogs. In some cases, there are specific requirements. An example is when a saw mill orders a specific length and/or diameter and quality. Logs from each assortment are put in a pile adjacent to a forest road. Each harvest area is defined by a geographical node. As there are several assortments at each harvest area, there are several piles and therefore we define a *supply point* as a combination of a geographical node and an assortment. Information connected to each supply point is a geographical location, assortment, and a volume. Production may take place for several days or weeks at a harvest area and logs are continuously transported to mills.

Once the harvesting is finished there is a need to empty the area within a certain time as the quality of the logs decreases with time. The emptying of an area can be controlled by imposing a cost or penalty for not removing the logs. A harvest area is often available all day around. However, trucks without a crane need a loader for the loading. A loader requires staff and is available within given working hours. A harvest area therefore has two time windows. The general availability (for all trucks) and the loader availability.

Demand and assortment groups

A *demand point* is defined as a customer order at an industry i.e. saw-, pulp- or papermill. A customer order is in turn defined by an *assortment group* and a volume. An assortment group can be one or several assortments. This means, for example, that both Spruce and Pine pulplogs can be used to satisfy a demand. In case there are limits on the proportions of different assortments in an order, this could be split up into several demand points. For example, if an order is for 1000 tons and it needs to be at least 30% of Spruce and Pine each, then we define three demand points; One for 300 tons of Spruce, one for 300 tons of Pine and one for 400 tons of combined Spruce and Pine. There are given opening hours at each demand point. If it is a paper or pulp mill, the time window is often very wide, often 24 hours per day, but for a small sawmill it may only be a few hours. The demand is often given on a weekly basis but broken down into minimum and maximum accumulated volumes during each day. This is illustrated in figure 2 where the accumulated demand is given for five days.

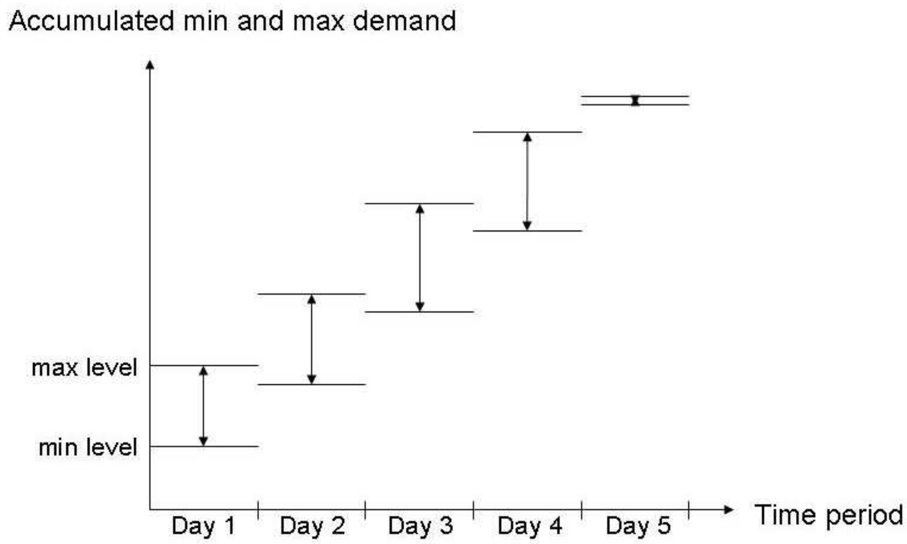

Figure 2: Example of a demand profile over five days.

Trucks and drivers

There are two types of logging trucks; with or without a crane. Figure 3 illustrates the two types of trucks. With a crane there is no need for a loader at the supply point. The loading capacity without a crane is about 40 tons and with a crane, three tons less i.e. 37 tons. Trucks belong to a haulier that owns one or several trucks. The working time for a truck is determined by the number of drivers during the day. A truck with three drivers can operate 24 hours each day whereas a truck with one driver is limited to about 10 hours. In the case of several drivers, they change at specified change-over locations. Each truck is located at a home base from where it starts and ends each day. In general we have separate costs for loaded and unloaded driving and working hours. The working hours are specified for each truck in detailed schedules.

Figure 3: Examples of a standard truck without a crane (left) and with a crane (right).

Distances and geographical nodes

There are four different types of geographical nodes: supply points, demand points, change-over nodes and home bases. In figure 4, we have a map from a case study with related nodes. An important aspect of the geographical nodes is the possibility to compute distances and travelling times between all pairs of nodes. We make use of the Swedish national road data base NVDB, which has detailed information about all the roads in Sweden.

Objective and costs

The objective is to find the most efficient plan for the entire fleet of trucks. In our case we want to find the minimum cost at which demand can be satisfied. In order to obtain a model which is both a flexible and a robust we have included a set of costs and priorities for the different elements. The most obvious cost is the actual routing cost. This cost is defined by a unit distance cost for the loaded and unloaded distance travelled. In addition, we have a cost associated with the working time, that is, the time the truck is in operation. There may be situations when it is not possible to satisfy the demand. Then we have included a unit volume penalty for not satisfying the demand. This typically represents the cost of buying the assortments on the wood market. Often there is a need to clear a harvest area and in order to control this, we have a unit

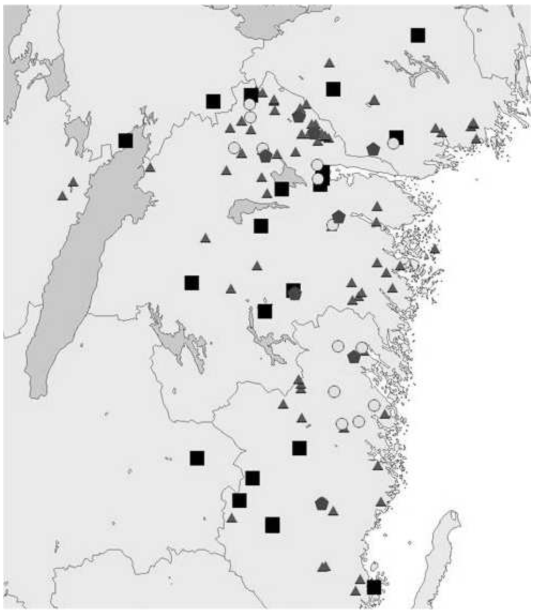

Figure 4: A map from a case study showing the distribution of geographical nodes. (demand points: squares, supply points: triangles, home base: pentagones, change-over nodes: circles)

volume penalty for each volume still left after the planning period. This is related to the quality deterioration of the logs and to administration costs.

3 Solution metods

The focus of this article is to develop a solution method that can be used in a DSS to solve the logging truck problem for large instances in a short time. The development of the system RuttOpt has been ongoing during 2003-2006. This is also true for the development of planning methods. In Palmgren et al. [18] and Palmgren et al. [19] Branch & Price (B&P) methods are used to solve a column (route) based formulation of an easier problem with a one day planning horizon and one truck type. The subproblem for finding routes was based on various heuristics. The approach works for smaller instances and shows large savings in comparison with manual solutions. In Palmgren [17], a modified subproblem was formulated and tested with B&P on a one day case. This is based on the smallest case study with one time period in section 4.3. The B&P approach however, fails, even with long solution times, to find feasible solutions. This motivates the need to develop more efficient and reliable solution methods.

3.1 Solution approach

The approach taken in this paper is to use well known and fast methods in a two-phase hybrid heuristic. In the first phase, we solve an LP model which is a relaxed and simplified version of an IP formulation of the full problem. It considers the flow between supply and demand points for individual trucks and days. Restrictions are on supply, demand and time availability of trucks. This model is more detailed than the traditional tactical destination models as it includes decisions about trucks. With the LP solution as a basis, we can form *transport nodes* which are full loads picked up at one or several supply points and delivered to one demand point. With this first phase we have transformed the problem into a vehicle routing problem with time windows (VRPTW). Then we make use of the unified tabu search algorithm (UTSA) by Cordeau et al. [5] that is developed for the VRPTW. The UTSA uses the transport nodes as customers to describe the composition of a route. We use an extended version of UTSA, called EUTSA, where we have added some features not included in the original method. This include handling differences in supply and demand and multiple home bases. After using EUTSA for a specified computational time, we remove some transport nodes and generate some new ones and reuse EUTSA as long as there is any solution time left within the allowed limit. The hybrid algorithm is summarized below.

Phase 1. Generate transport nodes

- **1a)** Solve an LP problem (a relaxed routing problem).
- **1b)** Form transport nodes based on the LP-solution.

Phase 2. Routing of transport nodes

- **2a)** Find an initial solution.
- **2b)** Apply EUTSA to solve VRPTW.
- **2c)** Update transport nodes.
- **2d)** If CPU time is available, go to step 2b). Otherwise stop.

In the remainder of the paper, we use "vehicle" instead of "truck" as this is the standard for VRP in the literature. The approach can be described with a simple example using only one vehicle. In figure 5 we solve a flow problem and get a solution (left part). Given this solution, we first decide the full truckloads going from one supply point to one demand point. We assume that a full truckload has a given weight or volume, e.g., 40 tons. We can identify three such, given the LP solution; they are denoted A, B and C. We can also find a fourth, denoted D, by combining two supply points. We have now established four transport nodes. In figure 6, we apply EUTSA to find a route which starts and ends at a home base and includes all the transport nodes. Each node includes a service time and has different starting and ending locations. In the same figure we also give the full route with numbers indicating the order in which the vehicle drives between nodes.

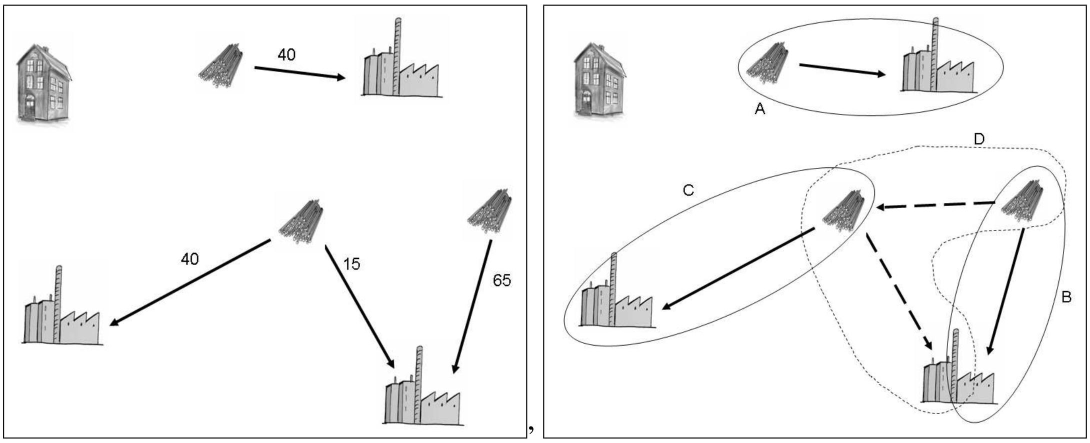

Figure 5: Illustration of the first phase of the hybrid method. Left: a flow solution from the LP problem. Right: Formation of four transport nodes (A-D).

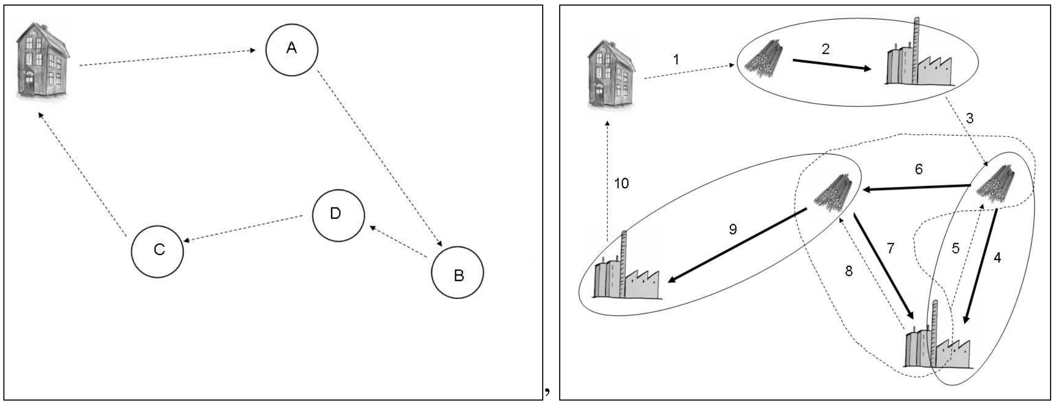

Figure 6: Illustration of the second phase of the hybrid method. Left: a route defined by the four transport nodes (Here the nodes have no geographical meaning.) Right: The actual physical route.

3.2 Phase 1: Generation of transport nodes

LP model

A relaxation of the full routing problem is when we deal with individual truck flow between any pair of supply and demand points. The variables in this formulation are defined as

x_{ijvt}	=	flow from supply point i to demand point j using vehicle v in period t
ℓ_{it}	=	storage at supply point i at the end of time period t (t=0 indicate initial supply)
h^+	=	total time to perform all transportations

Each supply point consists of a given assortment and each demand point a given assortment group. Supply can be planned to increase during the time periods due to harvesting. However, in most planning situations, the inventory known in the first time period is used throughout the entire planning period. Given the possible mix of assortments in the assortment groups and the given destinations, we have a set of possible combinations between supply and demand points. There could also be other restrictions, e.g. agreements or ownership, that state that some vehicles are not allowed to visit certain supply or demand nodes. We refer these combinations to the set W_v for each vehicle v. There is a standard assumption when the cost is calculated in tactical planning that the vehicle drives back and forward between supply and demand points. It is important to include the time capacity for each vehicle as this will effect the construction of routes in phase 2. Otherwise, there may, for example, be a case where all flows use supply points without a loader. Then, no transport nodes are constructed for vehicles without a crane and no routes can be found for these vehicles.

The index set and parameters of the LP model are defined as follows.

V	:	set of vehicles
T	:	set of time periods
I	:	set of supply points
J	:	set of demand points
s_{it}	:	additional supply at supply point i in period t
d^-_{jt}	:	accumulated lower demand at demand point j in period t
d^+_{jt}	:	accumulated upper demand at demand point j in period t
c_{ijv}	:	unit transportation cost between supply point i and demand point j using vehicle v
u_i	:	bonus for loading one ton at supply point i
v_j	:	bonus for unloading one ton at demand point j
f_{ij}	:	unit transportation (and loading/unloading) time between supply point i and demand point j
W_v	:	set of possible links between supply and demand nodes for vehicle v
h_{vt}	:	% of total transportation time (h^+) vehicle v is allowed to utilize in period t
M_{jt}	:	penalty for each ton of unfulfilled demand j in period t

The LP problem with time periods can be formulated as

$$
\begin{array}{llrcll}
[P1] & \min \ z = & \sum_{t\in T}\sum_{v\in V}\sum_{(ij)\in W_v}(c_{ijv}-u_i-v_j)x_{ijvt}+\sum_{j\in J}\sum_{t\in T}M_{jt}s_{jt} & & & \\
\text{s.t.} & & \ell_{i,t-1}+s_{it}-\sum_{v\in V}\sum_{(ij)\in W_v}x_{ijvt} & = & \ell_{it}, & \forall i\in I, t\in T \quad (1)\\
& & \sum_{v\in V}\sum_{t\in T, t\le p}\sum_{(ij)\in W_v}x_{ijvt}+s_{jp} & \ge & d^-_{jt}, & \forall j\in J, p\in T \quad (2)\\
& & \sum_{v\in V}\sum_{t\in T, t\le p}\sum_{(ij)\in W_v}x_{ijvt} & \le & d^+_{jt}, & \forall j\in J, p\in T \quad (3)\\
& & \sum_{t\in T}\sum_{v\in V}\sum_{(ij)\in W_v}f_{ij}x_{ijvt} & = & h^+, & \quad (4)\\
& & \sum_{(ij)\in W_v}f_{ij}x_{ijvt} & \le & h_{vt}h^+, & \forall v\in V, t\in T \quad (5)\\
& & x_{ijvt} & \ge & 0, & \forall v\in V, (ij)\in W_v, t\in T \quad (6)\\
& & \ell_{it} & \ge & 0, & \forall i\in I, t\in T \quad (7)
\end{array}
$$

Constraint (1) state the supply and integration between time periods. Constraints (2) and (3) state the accumulated lower and upper bound of the demand respectively. The penalty M_{jt} is high to ensure fulfillment of the demand if possible. Constraints (4) and (5) provide the time capacity for each truck and time period. It is used to force all vehicles to have similar work levels in relation to the available time. Constraints (6) and (7) are the non-negative restrictions on the variables. The objective is to minimize the overall transportation cost including potential penalties and bonuses.

Once problem [P1] is solved we will use the solution to construct transport nodes. However, in order to introduce extra flexibility we resolve problem [P1] once again, but with another demand and with the supply reduced in accordance with the first solution. The new demand in constraint (2) is set as $d^-_{jt} := \gamma * d^-_{jt}$, and in constraint (3) as $d^+_{jt} := d^+_{jt} - \sum_{v\in V}\sum_{t\in T, t\le p}\sum_{(ij)\in W_v} x^1_{ijvt}$, and the supply $s_{it} := s_{it} - \sum_{v\in V}\sum_{(ij)\in W_v} x^1_{ijvt}$, where x^1_{ijvt} is the first solution. The new solution is denoted x^2_{ijvt}. Typical values of γ are in the range 0.05-0.20.

Generation of transport nodes

First we compute the flow between all pairs of supply and demand points given the solution from [P1]. The flow is denoted f_{ij}, where $f_{ij} = \sum_{v\in V}\sum_{t\in T}(x^1_{ijvt} + x^2_{ijvt})$. Given the flow we form transport nodes in two steps. In step 1 we identify all the transport nodes that represent a full truckload between one supply point and one demand point. We construct $\left\lfloor \frac{f_{ij}}{l_v} \right\rfloor$ transport nodes where l_v is the basic vehicle load, e.g. 40 metric tons. The remaining flow that is used in the LP solution but that has not been allocated to a transport node yet is $f^s_{ij} = f_{ij} - \left\lfloor \frac{f_{ij}}{l_v} \right\rfloor * l_v$. In step

2 we take all supply points i with a remaining flow $f^s_{ij} > 0$, that is now less than a truckload, and combine these with other supply points to form full truckloads. We use two approaches to form transport nodes from $f^s_{ij} > 0$.

The first approach for combining small piles to make a full truckload is to start with the largest remaining flow $f^s_{ij} > 0$. If enough volume exists to fill a truck at the supply point i and nearby supply points that can be used to fulfill demand j, a new transport node is generated. Then the next remaining flow $f^s_{ij} > 0$ is used to try to create a transport node, and so on until either all demand is fulfilled or all flow $f^s_{ij} > 0$ has been utilized. In this process, we may use an interval, say 35-40, tons to define a full truckload. The reason is that we want to include the experience that the information of the volumes in small piles often is inaccurate and this provides some flexibility and is a good approximation of how it is done when the routes are planned manually. Each added supply point is chosen as the supply point which has supply left and which increases the distance traveled (within the transport node, inserted at the cheapest position within the transport node) the least. This also gives the order of the supply points within the transport node. An example of the process is illustrated in figure 7. The supply quantity that is left at each supply point is given above the pile in the figure and the flows f^s_{ij} are given above the arcs. In the example, there are three flows $f^s_{ij} > 0$. The highest one is from supply point A with a flow of 20 tons. It is not a full truckload so another supply point has to be included in the transport node. Supply point B increases the driving distance the least, giving in total 30 tons to be picked up at the points A and B. Next, supply point C is added since this increases the cost for the whole transport node the least. If ten tons are picked up at point C then a full truckload is generated and hence we create a transport node with the supply points C-A-B and then to the demand point. Next, supply point E is chosen since $f^s_{ij} = 15$ is the second largest one. Since the supply point has 65 tons left, a full truckload can be loaded at this point and hence a transport node is created with supply point E to the demand point. Now, enough transport nodes have been added to fulfill the demand.

The second approach is to formulate a mixed integer programming (MIP) model. This model also determines the order in which to visit the different supply points in the most cheaply way and how much to load at each of the supply points. In the model, we use different levels to represent the potentially different piles that are combined within the transport nodes. The parameters used are

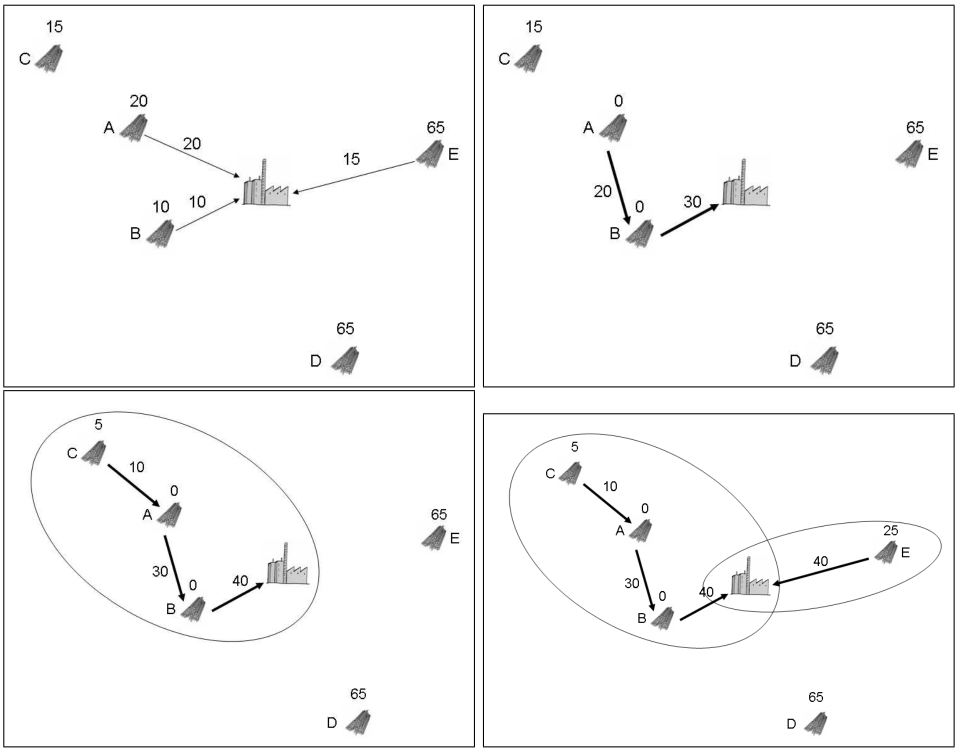

Figure 7: Illustration of the first approach to construct two transport nodes from small piles.

I^s	:	set of supply points with supply left after deduction for the created transport nodes in step 1.
J^s	:	set of demand points with demand left after deduction for the created transport nodes in step 1.
N	=	$I^s \cup J^s$
L	:	set of supply levels, with L_0 as the first level, L_n as the last (final) level, and L_m as all other levels
L_l^b	:	the level before level l
B_0	:	set of arcs (with index i,j,l) between nodes in first and second level, where $i \in I^s, j \in I^s$, and $l \in L_0$
B_m	:	set of arcs (with index i,j,l) between nodes in all intermediate levels, where $i \in I^s, j \in I^s$, and $l \in L_m$
B_n	:	set of arcs (with index i,j,l) between nodes in the last level and the demand points, where $i \in I^s, j \in J^s$, and $l \in L_n$
B	=	$B_0 \cup B_m \cup B_n$
s_i^s	:	remaining supply at node i, where $i \in I^s$
d_j^s	:	remaining demand at node j, where $j \in J^s$
c_{ij}^s	:	cost of using arc from node i to nod j where $i \in I^s$ and $j \in N$
c_ϵ	:	small cost for flow on an arc
M^+	:	upper bound on one truckload
M_l	:	lower bound on one truck load (if the arc is used)

We use the following variables

y^s_{ijl} = flow from node i to node j, starting at level l where $ijl \in B$
z^s_{ijl} = number of truck loads (integer) to be transported on the arc from node i to node j starting at level l

The problem can be formulated as

$$
\begin{array}{llllll}
[P2] & \min z = \displaystyle\sum_{(ijl)\in B} (c^s_{ij} z^s_{ijl} + c_\epsilon y^s_{ijl}) & & & & \\
\text{s.t.} & \displaystyle\sum_{(jl):(ijl)\in B_0} y^s_{ijl} & \leq & s^s_i, & \forall i \in I^s & (8) \\
& \displaystyle\sum_{(il):(ijl)\in B_n} y^s_{ijl} & \geq & d^s_j, & \forall j \in J^s & (9) \\
& \displaystyle\sum_{j\in I^s, l'\in L^b_l:(jil')\in B} y^s_{jil'} & = & \displaystyle\sum_{j\in N:(ijl)\in B} y^s_{ijl}, & \forall i \in I^s, l \in L_m \cup L_n, & (10) \\
& \displaystyle\sum_{j\in I^s, l'\in L^b_l:(jil')\in B} z^s_{jil'} & = & \displaystyle\sum_{j\in N:(ijl)\in B} z^s_{ijl}, & \forall i \in I^s, l \in L_m \cup L_n, & (11) \\
& y^s_{ijl} & \leq & M^+ z^s_{ijl}, & \forall (ijl) \in B & (12) \\
& y^s_{ijl} & \geq & M^+ z^s_{ijl} - M^+ + M_l, & \forall (ijl) \in B_n & (13) \\
& y^s_{ijl} & \geq & 0, & \forall (ijl) \in B & (14) \\
& z^s_{ijl} & & integer. & \forall (ijl) \in B & (15)
\end{array}
$$

Constraint (8) state the supply and constraint (9) the demand. We note that these values are what remains after the direct and full truckloads have been generated. Constraints (10) state the node balance flow and (11) the node balance in integer truckloads. Constraints (12) and (13) provide the link between truckloads and flow. Constraint (13) force the flow from the final supply point in a transport node to a demand point to be full truckloads (M^+) except for the last load which has to be a load of at least M_l tons. Constraint (14) are the non-negativity constraints on the y-variables and (15) the integer restrictions on truckloads. The objective is to minimize the cost of designing the truckloads.

The formulation can be illustrated by figure 8. At level 0, the nodes are supply nodes with strength $+s^s_i$. At all other levels, the supply nodes are intermediate nodes and the demand nodes have a strength $-d^s_j$. Flow from one level to the next between different supply points corresponds to one or more vehicles (given by z^s_{ijl}) loading at both supply points, starting with the one at the lower level. Flow y^s_{ijl} where $l \in L_f$ corresponds to loading less than or equal to y^s_{ijl} tons (depending on whether other supply points are visited before i) and delivering y^s_{ijl} tons to demand point j.

The nonnegative arc cost c^s_{ij} in [P2] is the cost for driving a vehicle from node i to node j with $c^s_{ii} = 0$. When $j = J^s$, a constant is added to the cost which corresponds to an average cost of driving a vehicle between two transport nodes. This is to resemble the actual cost for each transport node when it is included in a route and hence to avoid generating several transport nodes with low total loaded volume. A small cost, c_ϵ, is added to each flow to avoid some potential multiple optimal solutions where the only difference between solutions is an excess

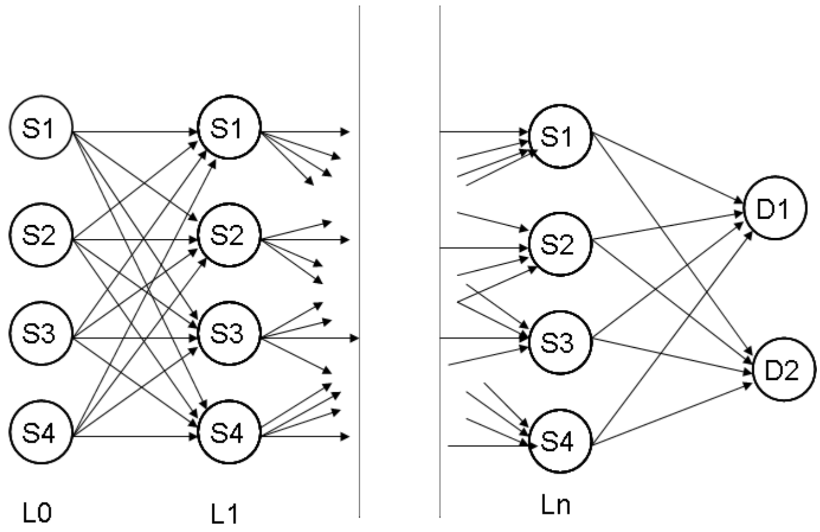

Figure 8: Illustration of the underlying network for problem [P2].

flow on some arcs. There are restrictions on the supply and demand points and node balance on the flow (x^s_{ijl}). The flow on an arc is limited to the number of trucks the arc is open for. If an arc is open for a number of trucks, then the flow must be at least M_l tons for the last truck and a full truckload for all others. We also have a node balance on the number of truckloads. This is to ensure the solution describes a number of tours where each tour goes from one or more supply points to one demand point.

The transport nodes are generated by following the flow and truckloads from the demand nodes J^s at the last level (L_n) back through the different levels to the first level (L_0). One transport node is generated for each truckload entering a demand point. For example if $z^s_{ijl} = 5$, where $l \in L_f$, then 5 transport nodes are generated to demand point j from supply point i (some or all of them possibly visiting one or more supply points before supply point i). For each truckload, the different supply nodes that are visited on the way back to level 0 are included in the transport node and the flow can easily be used to determine the loading volume at each supply node.

Information about transport nodes

We will use the transport nodes as customers in the VRPTW formulation. The transport nodes have additional information that it is important to consider. Each transport node has a time window within which a vehicle is allowed to visit the node and also a service time. Since a transport node is made up of both loading at supply points and unloading at demand points the service time consists of

- the time it takes to load a vehicle at the supply point(s)
- the time it takes to drive between the supply point(s) and the demand point with a loaded vehicle

- the time it takes to unload a vehicle at the demand point
- the time a vehicle has to wait before loading or unloading capacity is available.

The time windows are used to make sure a vehicle only visits supply and demand points when they are open. At some supply points, there are loaders that can be used which makes the time to load and/or unload less than if a vehicle performs it itself. A loader has limited working hours which leads to different service times depending on what time of the day it is used. This is handled by using different time windows with different service times depending on when each node is visited.

Since some vehicles do not have a crane, they have to rely on a loader to load and unload the vehicle. Therefore, for each transport node, two different sets of time windows are generated, one set for vehicles with a crane and one set for vehicles without a crane. Vehicles with a crane can make use of a loader as well if the loader is present when the loading starts. Therefore, the duration of a visit for a transport node depends on if a loader is present also for a vehicle with a crane.

A transport node is in general not a geographical point but at least two, a pickup point and a delivery point. This is handled by using an asymmetric distance matrix where the distances are measured from the last visited position in a transport node to the first visited position in the next transport node. For example, suppose that we have two transport nodes, n_t^1 and n_t^2. They have pickup points at p_1 and p_2 and delivery points at d_1 and d_2, respectively. The distance from node n_t^1 to node n_t^2 is $dist(d_1, p_2)$ where $dist(a, b)$ is the distance from point a to point b. Similarly, the distance from node n_t^2 to node n_t^1 is $dist(d_2, p_1)$.

Any required breaks and change of drivers for vehicles are also handled with the use of transport nodes. In this case, the starting position for the transport node will most often be the same as the finishing position. A time window and a service time is added to make sure the break or change of drivers is performed at the right point in time and that it lasts for a specific duration. These transport nodes need only to be visited by the vehicle where the break or change of drivers is necessary.

3.3 Phase 2: Routing of transport nodes

Phase 2 consists of three steps. First we establish a feasible solution and then apply the extended version of UTSA (Cordeau et al. [5]). As a part of phase 2, we update the transport nodes and resolve the updated problem. In this section we discuss and motivate these steps.

Routing methodology

The VRPTW for which UTSA has been developed, is defined to find a route for each of the m vehicles in the planning problem. The routes are defined on a graph $G = (V, A)$, where

$V = v_0, v_1, \ldots, v_n$ is the vertex set and $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ is the arc set. Vertex v_0 represents a depot at which the fleet of vehicles is positioned. Associated with the remaining vertices of V are a nonnegative load q_i (with $q_0 = 0$), a nonnegative service duration d_i (with $d_0 = 0$), and a time window $[e_i, l_i]$, where e_i and l_i are nonnegative integers. Each arc (v_i, v_j) has an associated nonnegative cost c_{ij}. The load and time capacity of the vehicle k is Q_k and T_k, respectively. The VRPTW consists of designing m vehicle routes on G such that:

- every route starts and ends at the depot.
- every customer (or transport node) belongs to exactly one route.
- the total load and duration of route k does not exceed Q_k and T_k respectively.
- the service at customer i begins in the interval $[e_i, l_i]$, and every vehicle leaves the depot and returns to the depot in the interval $[e_0, l_0]$.
- the total cost of all vehicles is minimized.

Using transport nodes ensures that the capacity of the vehicle cannot be exceeded. We have that $q_i = 1$ for all customers. Also, the time capacity is described through time intervals representing working hours. An important feature of UTSA is the possibility of exploring infeasible solutions. During the solution process, a set of penalty weights representing load, duration and a time window constraint are dynamically updated.

Since there is no requirement to visit all customers (i.e. transport nodes) but only enough to fulfill all orders the UTSA can not be used directly to solve our problem. To be able to handle transport nodes not used, we introduce a *virtual vehicle*. The transport nodes that are handled by the virtual vehicle do not induce any transport costs and the time windows and service times are relaxed. To ensure that all orders are fulfilled, a penalty cost is added for each ton of an order that is not fulfilled (each ton below the lower bound or above the upper bound). The UTSA code has been changed to be able to handle the features outlined below. With these changes we denote the new algorithm *the extended unified tabu search algorithm* (EUTSA).

Vehicle - node restriction Limits the nodes which the different vehicles are allowed to visit. This reduces the neighbourhood of a solution in the tabu search method.

Location of a customer/ transport node A transport node has different start and finish locations. This is handled within the cost and the distance matrices between customers.

Multiple depots Different vehicles can have different starting nodes and finishing nodes. This is handled with the distance matrices. IN EUTSA we consider different distances from the different starting nodes to the different customers and from the different customers to the different finishing nodes for each vehicle.

Multiple time windows More than one time window (potentially with different service times) may be valid for customers.

Demand over multiple periods Lower and upper limits on demand, which are accumulated over the time periods. The total accumulated weight in time period t transported to fulfill demand point j is denoted d^T_{jt}. Assume that a transport node corresponding to delivering l_v tons to demand point $\hat{j}$ is added to vehicle $\hat{v}$ in time period $\hat{t}$. Then all accumulated weights $d^T_{\hat{j}t'}$ with $t' \in T$ where $t' \geq \hat{t}$ are updated with l_v tons. This updating is handled by adding a penalty for each ton that is below the lower bound or above the upper bound on the demand for each time period. If a transport node is removed from a vehicle, then the relevant d^t_{jt} are updated accordingly and the potential penalties are included in the cost evaluation of removing the transport node.

Different full truckloads Different vehicles can load different volumes depending on whether or not they have a crane. Vehicles without a crane just operate from large supplies with loaders. When a full truckload is generated the weight is, say 40 tons, at supply points with a loader and, say 37 tons, in other nodes. This information follows each truckload or transport node.

Working schedules Different vehicles have different working times. This is handled by only adding transport nodes after a vehicles starting time each day and by penalizing any time used after a vehicle's finishing time.

Initial solution

If an initial solution is not available, it is created as follows. Initiate m routes with a *free time* for each route which is the amount of time the vehicle in the route is standing still with nothing to do. The initial free time for route k (to be driven by vehicle k) is set to (finishing time - starting time) for vehicle k, with the free time for the virtual vehicle $k_v = 0$. Sort the transport nodes according to how many different vehicles are allowed to visit them with the transport node with the least number of allowed different vehicles first. We define $c(i, k)$ as the minimal cost for visiting transport node i with vehicle k. If vehicle k cannot fulfill all the time windows while visiting transport node i, the cost $c(i, k) = 2 * M$, where M is more than the highest possible cost of a feasible route. If transport node i is visited by the virtual vehicle k_v and the order that transport node i is used for is not fulfilled, the cost $c(i, k_v) = M$, otherwise the cost $c(i, k_v) = 0$. For each transport node i:

(i) Determine $c_l(i) = \min\{c(i, k) : k \in K\}$.

(ii) Sort the routes in a sequence where the route with most *free time* is earliest.

(iii) Transport node i is added to the first route k^r in the sequence with $c(i, k) \leq 1.5 * c_l(i)$.

(iv) Update the free time for route k^r.

Repeated solving of the EUTSA

When we have applied EUTSA to VRPTW with a given computational time, we update the current set of transport nodes. Some are removed and some new are added. Then we reapply EUTSA. The reason for this is that we cannot guarantee an optimal set of transport nodes from phase 1. Therefore we make local changes to the set in order to improve the solution quality. Transport nodes that are not used in the current solution of the EUTSA are removed and new transport nodes that potentially can improve the quality of the solution are identified and added to the problem after analyzing the current solution. From phase 1, we have a surplus of nodes, say 5%, and these can be exchanged with a new set. The new set is constructed using two simple rules.

Rule 1 is to make sure that (if possible) each demand point has at least two more transport nodes than are needed to satisfy the lower limit of the demand. New transport nodes are generated by simply using the closest supply point(s), not emptied, to construct full truckloads. This is illustrated in figure 9 where we have four transport nodes corresponding to S1-D1, S2-D1, S3-D1 and S4-D1. Suppose the demand at D1 is for five full truckloads, then we can generate an additional three by taking the closest two supply points not used, assuming that there is enough supply for one truckload in E3 and for two full truckloads in E2. In this example, we would generate one transport node corresponding to E3-D1 and two transport nodes corresponding to E2-D1.

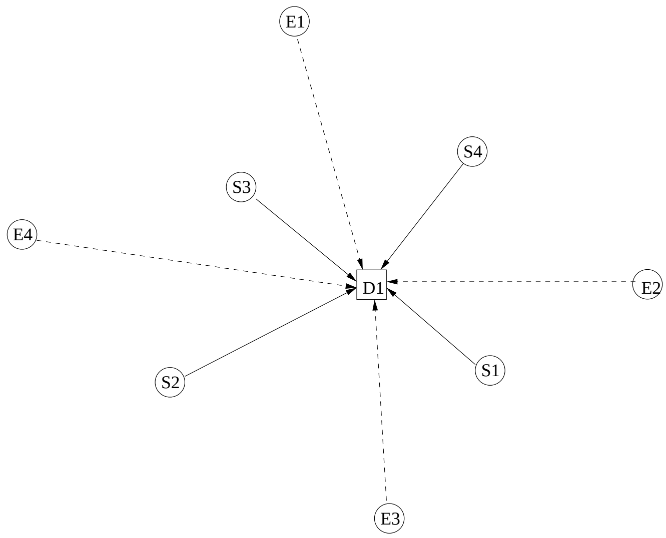

Figure 9: Illustration of rule 1.

Rule 2 is to achieve more effective routes. Given a vehicle and a transport node, we check if the supply can be exchanged for another which is better located. This is illustrated in figure 10, where a route for a vehicle is given in solid lines. The three transport nodes represent the transports S1-D1, S2-D2 and S3-D3. Supply points E1 and E2 were not included in the LP solution to construct transport nodes but they are evaluated given the actual route. From this we can identify that constructing a new transport node for E2-D2 would be beneficial as this would reduce the distance traveled.

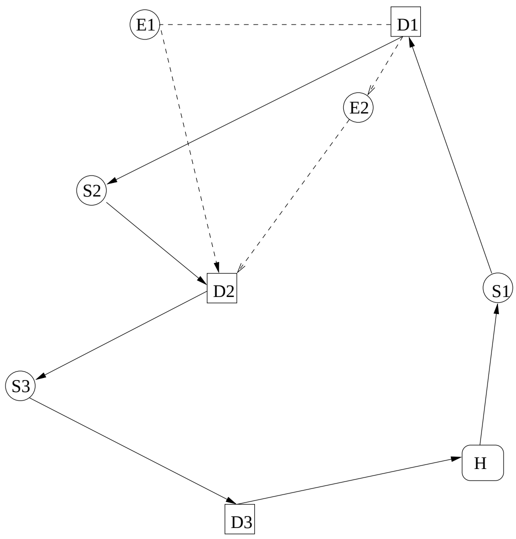

Figure 10: Illustration of rule 2.

The number of times that the transport nodes are updated and that EUTSA is used depends on the available solution time. The user sets a maximum time which is used as convergence criteria.

4 Numerical results

4.1 System

We have used the system RuttOpt for all experiments. The main components of the system are given in figure 11. The system uses the Swedish national road database (NVDB) with detailed information on all roads, a geographical user interface (ESRI ArcView), a database (Microsoft Access) with all case information, routines for report generation and an external route planner communicated through a defined interface.

Different solution methods can be used in the route planner as the interface is defined by a set of input/output files. The proposed solution method (route planner) is developed in C and uses as its basis the UTSA described in [5]. All experiments have been performed on a standard PC with a Pentium 4, 2.4 GHz processor and 1 GB internal memory.

Reports that can be generated from the user interface. One example is Gantt schemes, see figure 12.

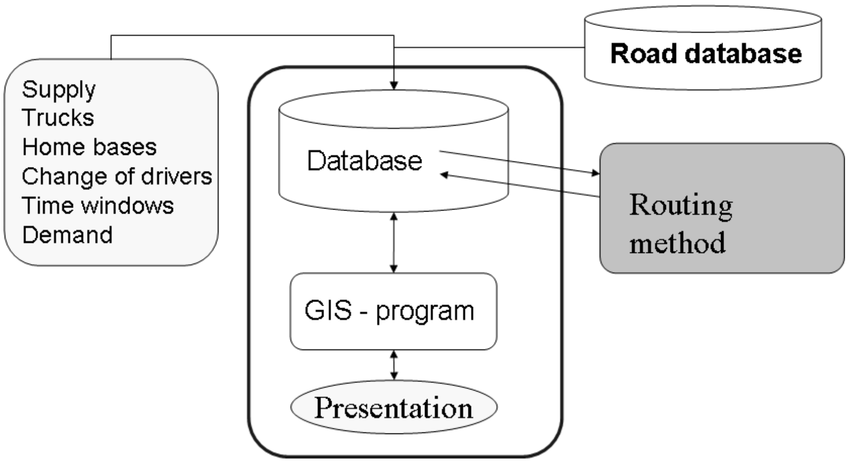

Figure 11: Overview of the RuttOpt system.

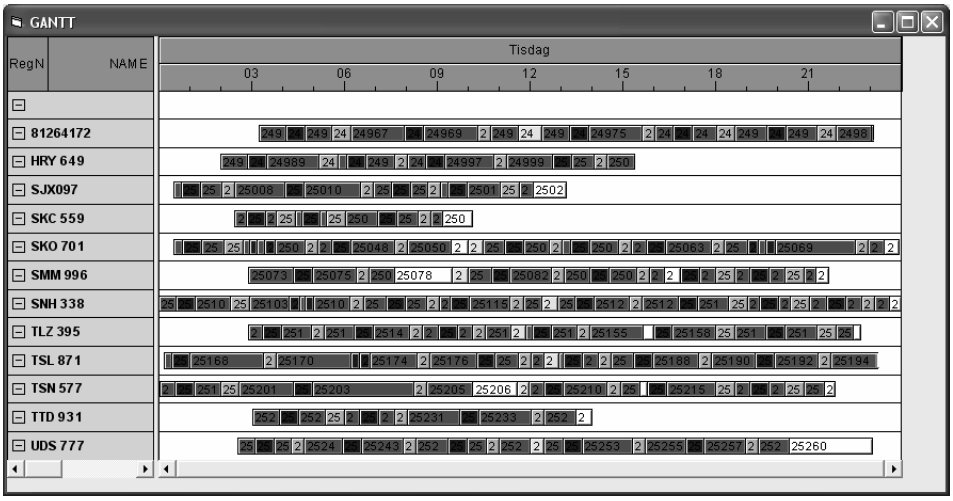

Figure 12: A Gantt scheme for one day from case H. Different colors represent different actions. For example, empty run, loading, loaded run, unloading, change of drivers, and breaks.

4.2 Case studies

We have used two case studies as the basis for our experiments. The first is based on data collected at the forest company Holmen Skog and the second using data from the forest company Stora Enso and three haulage companies. The purpose of the first case was to develop and test the solution approach and compare this with manual results covering the three days. In this case, each vehicle kept a diary for all the transports carried out during three days. The available road side inventory at the beginning of day one was recorded and used as overall supply. The second case is used to test the performance of the proposed solution method for a large scale problem. In this case, data for five days was collected in co-operation with Skogsåkarna, a large hauling association. Skogsåkarna is responsible for almost all the transports of roundwood in the area concerned. Information on the size of the case studies, called H and SE, is given in table 2.

Case	Holmen (H)	Stora Enso (SE)
# vehicles	12	110
# industries	22	74
# demand points	24	113
# supply points	410	2,531
# time periods	3	5
demand volume (tons)	7,511	101,018
supply volume (tons)	33,331	261,260

Table 2: Information about the two base case studies.

Given the two base cases, we have extracted a set of instances to test the performance of the solution approach. In table 3 we give the instances from case H and their characteristics. The column "Limited volume" indicates if the available supply points and their volume is exactly the same as the supply points that were visited and the volumes loaded in the manual solution, i.e. the supplied volume for these cases is about the same as the demand volume. The column "Fixed trucks" refers to whether the trucks are restricted to visiting the same supply points as they did in the manual solution. Case H7 has more small piles than the others. The instances for the case SE are given in table 4.

Case	No. Time periods	Limited Volume	Fixed truck	Data
H1	1	Yes	No	Day 1
H2	1	Yes	No	Day 2
H3	1	Yes	No	Day 3
H4	3	Yes	Yes	Day 1-3
H5	3	Yes	No	Day 1-3
H6	3	No	No	Day 1-3
H7	1	Yes	No	Day 1

Table 3: Information about the instances based on case H.

Case	No. Time periods	# trucks	Case	No. Time periods	# trucks
SE1	1	110	SE8	1	90
SE2	1	10	SE9	1	100
SE3	1	20	SE10	5	12
SE4	1	30	SE11	5	20
SE5	1	40	SE12	5	30
SE6	1	50	SE13	5	40
SE7	1	70	SE14	5	50

Table 4: Information about the instances based on case SE.

4.3 Numerical results

In the experiments we have aimed to test the following aspects of the solution method.

(i) Solution quality

(ii) Solution convergence wrt time

(iii) Impact of the LP model on constructing transport nodes

(iv) Impact of the method for combining smaller piles

(v) Impact of the starting solution

(vi) Solution convergence wrt the number of trucks and time periods

Case H has been used to test (i) and (ii) since this is the only one for which we have manual solutions. Case SE has been used for (ii)-(v). Here, we have constructed different instances by changing some input data of the original case and/or the parameters of the method.

Solution quality

In table 5 we give the results from instances H1-H6. Only one call to the EUTSA is used for the instances H1-H5 (since there is no excess from which supply to create additional transport nodes) and the solution time is 10 minutes. The column "# TN" gives the number of transport nodes used. In the manual solution, some vehicles were loaded late one day, drove home and were only unloaded the next day. These loads are included in the demand in the instances H4-H6 but not in the instances H1-H3 since these instances are generated from the explicit parts of the manual solution for each day. Therefore, the total demand for the instances over a three day period is more than the total demand for the instances H1-H3 added together. Even though there are restrictions, the EUTSA do find considerably better solutions than the manual solutions. In a practical setup, the freest case, i.e. H6, would be used. However, the improvement would in general be lower since attention has to be paid to the time periods after the ones studied here. Certain supply points might also have to be emptied (by setting u_i high for these supply points) since the maximum time from when a tree is harvested until it has to be transported away from the forest is 30 days. Only a small improvement is reached for instance H4. This instance is very restricted since the solution has to be very similar to the manual one (since the same vehicles have to visit the same supplies as in the manual solution and exactly the same supply points have to be used).

Case	Manual	EUTSA	# TN	Improvement (%)
H1	41503	39496	70	4.84
H2	38454	36744	78	4.45
H3	33697	30432	72	9.69
H4	118965	116780	232	1.84
H5	118965	109779	230	7.72
H6	118965	82210	301	30.90

Table 5: Results from cases H1-H6.

Solution convergence wrt time

In figure 13 we show the convergence behaviour of EUTSA when applied to case H1 and case H4. The manual solution is indicated with a horizontal line and the objective function values are scaled such that the manual solutions are represented by 100%. We can see that these solutions are directly better (i.e. the initial solution) than the manual ones and that the convergence occurs after about 4-5 minutes.

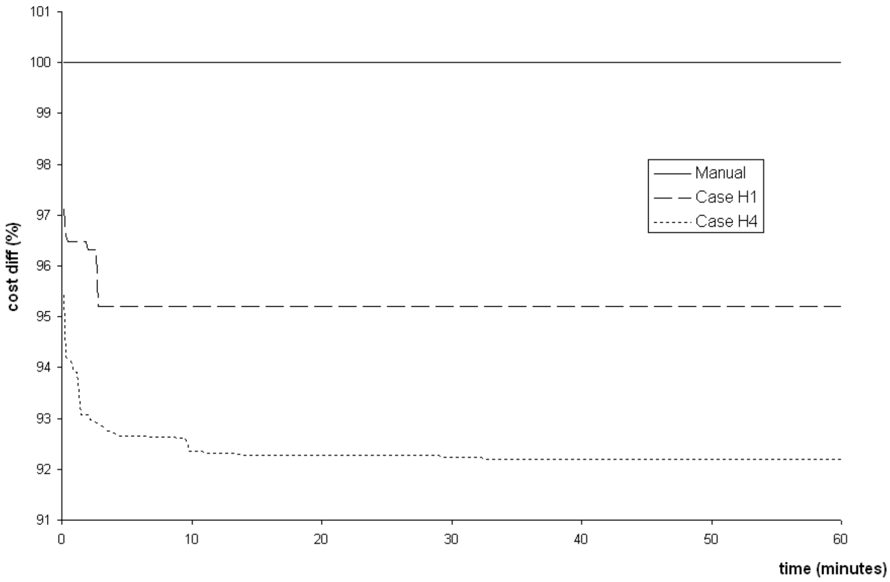

Figure 13: Convergence behaviour wrt to time for cases H1 and H4.

In case SE we do not have any manual solution, but figure 14 shows the convergence behaviour of EUTSA when applied to instance SE1. It is clear that this problem is quite difficult and when there is only one call to the EUTSA, the convergence is attained after a few hours. When multiple calls to the EUTSA are used (six in this case) the performance increases and so does the convergence time.

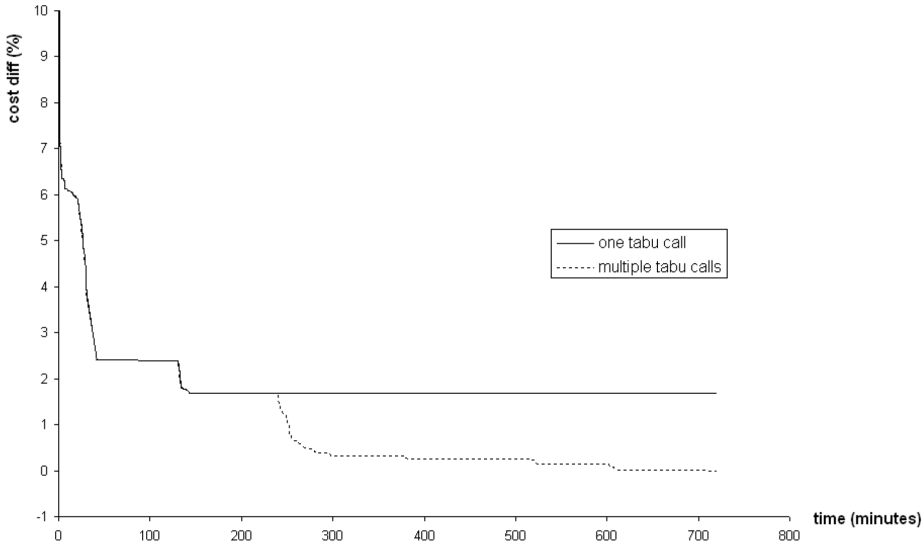

Figure 14: Convergence behaviour wrt to time for case SE1.

Impact of the LP model to construct transport nodes

Problem [P1] is first solved with the given bounds on demand. This problem is resolved with a new demand representing $\gamma\%$ of the given lower bound and with the supply and demand updated taking into account the first solution. The motivation is to generate some extra transport nodes for increased flexibility in the tabu search method. In figure 15, Instance H1 has been solved with different values of γ to generate more transport nodes. To make the comparison interesting, only one call to EUTSA is used. We observe that $\gamma = 10\%$ works well since higher values of γ do not decrease the cost much.

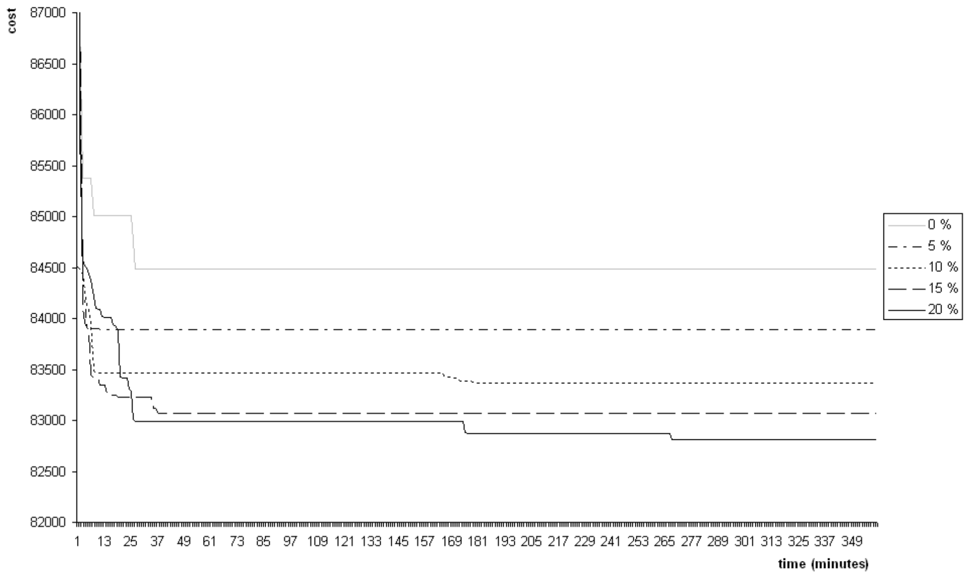

Figure 15: Comparison of solution quality for different values of γ.

Impact of the method of combining smaller piles

We have described two aggregation methods for combining smaller piles into a full truckload. The impact of these depends on the presence of small piles. This number of small piles will change between cases and planning situations. For example, in the summer it is important to remove small piles and then many piles needs to be included in the planning. In other situations, there may only be a small number of piles. In the tests, we have used the daily planning problems from case H and two instances from case SE. One case is constructed with more small piles in order to test the aggregation methods. In table 7 we show the results. The optimizing method works better when the supply and demand volumes are similar. This is because it is important to utilize more or less all the small piles to create transport nodes. If there is a considerable excess of supply volume, then there will be many small piles that are not needed. As a result many different options for combining small piles will work well when combining the routes and hence the quicker heuristic method will be just as good as the optimized.

Case	Heur	# TN	Opt	# TN
H1	40482	71	39496	70
H5	112076	232	109779	232
H6	82193	320 (278)	82210	301 (285)
H7	43515	70	42988	78

Table 6: Objective function values from using the heuristic and the optimized method to aggregate small piles into full truckloads. In case H6, two calls to EUTSA were used and the number within parentheses in column "# TN" is the number after the first call made.

Impact of the start solution

In table 7 we present a comparison of using different start solutions for the Tabu heuristic. To make the comparison interesting, only one call to EUTSA is performed for each run and a solution time of one hour is used. The start solution is generated using four different methods, ST1-ST4.

The first method (ST1) is the heuristic described earlier to find an initial solution to the VRPTW. Methods ST2 and ST3 are used for allocating each transport node to a randomly chosen vehicle. For ST2, the transport node is added at a randomly chosen position within the route. For ST3 the transport node is added at its cheapest position within the route. The fourth method (ST4) is to add the transport node randomly among all vehicles which give a feasible solution (i.e. the route fulfills all time windows) after the transport node has been added at its cheapest position within the route. No objective value is given when no feasible solution was found within the given solution time. For small cases the initial solution does not have an impact but for the larger it is important.

Case	ST1	ST2	ST3	ST4
H1	39496	39631	**39466**	39581
H6	**82249**	82946	83284	82437
SE2	30248	30172	**29422**	30452
SE3	58392	58648	58166	**58009**
SE4	**90301**	93415	93756	92866
SE5	122922	123320	122115	**120836**
SE6	**160376**	-	-	-
SE7	**198489**	-	198695	199570
SE8	**262348**	-	265643	-
SE9	**336105**	-	-	-

Table 7: Solution quality with different start solutions for the Tabu heuristic. The best solution is given in boldface.

Solution convergence wrt the number of trucks and time periods

When we use a large number of trucks and several time periods, the problem becomes very large. In such a case, the EUTSA gets a slow convergence. In order to improve the performance for the large instances, we have developed a simple approach where we use a rolling planning horizon. First we solve the problem for time period 1 and update supply and demand given the solution. This is repeated recursively until all the time periods are solved. For each time period t, problem [P1] is solved for all the time periods after and including t, but only the flow of period t is used when the transport nodes are generated.

In table 8, we present a comparison of the objective value of the solution when all time period are solved together (the basic method) and the objective of the solution when each time period is solved separately in the Tabu heuristic. The comparison is done both for Case H4 which has three time periods and instances SE10-SE14. A feasible solution for instances SE13 and SE14 is not found when the basic method is used with a solution time of one hour.

Case	basic	extension
H6	82210	83190
SE10	168847	172327
SE11	321291	333442
SE12	508740	509430
SE13	-	707409
SE14	-	898045

Table 8: Solution quality solving VRPTW for all time periods together compared to one time period at a time. No objective value is given when no feasible solution was found within the given solution time.

5 Concluding remarks

We have proposed a two-phase hybrid method to solve a very difficult and large routing problem. The main idea is to construct transport nodes which is a way to decompose the problem into a standard VRPTW problem. The transport nodes can be constructed by solving an LP problem. The method is implemented in a DSS and provides solutions in acceptable solution times. This method performs considerably better than manual solutions. Earlier methods for the application were unsuccessful in solving the large sized problems. The VRPTW is solved using a modified tabu search method. By introducing the concept of a virtual vehicle we can make a general VRPTW code applicable to situations where not all customers need to be visited.

The problem studied is an industrial application arising in forestry. There are many practical aspects and restrictions included in the problem that are hard to include in a general code. Some important aspects are therefore implemented as refinements to the solution obtained and add-on modules. At present, the EUTSA can not be used to determine if a change of drivers should be done when the vehicle is loaded or empty since it corresponds to performing operations within two different transport nodes at the same time (loading a vehicle is part of one transport node, changing drivers is another one, and then driving and unloading is part of the first node again). Therefore, this is handled separately between the calls to the EUTSA. We check if it would be cheaper to change the drivers during the loaded run just before the planned change of drivers or during the loaded run just after. If either of these are cheaper, then the transport node corresponding to the cheapest cost is changed to include a change of drivers as well with the consequence of a longer service time. This is because the service time is extended with the loaded run from the supply point to the location where the drivers change, the time it takes for the change and the loaded run to the demand point (or another supply point) and is deducted with the loaded run from the supply point to the demand point (or another supply point).

The EUTSA is not developed to determine if a vehicle should pickup a load at the end of one day, store it at the home base during the night, and deliver it at the beginning of the next day. This is handled in a post-solving manner. Given a solution, we check to see if it would be cheaper to pick up the next day's first delivery today. If this is the case, we lock this in the next call to the EUTSA.

The planning does not include queuing at supply or demand nodes. If the system is to be used in a planning environment where the solution is updated several times during the day or where queuing is an important restriction, this will be an important aspect to consider. An interesting future development is to include queuing directly in the EUTSA.

References

[1] Archetti C, Speranza M, Hertz A. A tabu search algorithm for the split delivery vehicle routing problem. Transportation Science 2006; 40(1): 64-73.

[2] Bent R, Van Hentenryck P. A twp-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. Computers & Operations Research 2006; 33: 875-893.

[3] Bräysy O, Gendreau M. Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. Transportation Science 2005; 39(1): 104-118.

[4] Cordeau J-F, Gendreau M, Laporte G, Potvin J-Y, Semet F. A guide to vehicle routing heuristics. Journal of the Operational Research Society 2002; 53: 512-522.

[5] Cordeau J-F, Laporte G, Mercier A. A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operational Research Society 2001; 52: 928-936.

[6] Crevier B, Cordeau J-F, Laporte G. The multi-depot vehicle routing problem with inter-depot routes. European Journal of Operational Research 2007; 176: 756-773.

[7] Epstein R, Morales R, Seron J, Traverso P, Weintraub A. A truck scheduling system improves efficiency in the forest industries, Interfaces 1996; 26(4): 1-12.

[8] Eriksson J, Rönnqvist M. Decision support system/tools: Transportation and route planning: Åkarweb - a web based planning system. In: Proceedings of the 2nd Forest Engineering Conference, Växjö, May 12-15, Sweden, 2003. p. 48-57.

[9] Forsberg M, Frisk M, Rönnqvist M. FlowOpt - a decision support tool for strategic and tactical transportation planning in forestry, International Journal of Forest Engineering 2005; 16: 101-114.

[10] Gendreau M, Laporte G, Potvin J-Y. Metaheuristics for the Capacitated VRP. In: Toth P, Vigo D, editors. The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, PA 2002; 129-154.

[11] Gingras C, Cordeau J-F, Laporte G. Un algorithme de minimisation du transport à vide appliqué a l'industrie forestière, Report CIRRELT, Montreal, Canada, 2006.

[12] Gronalt M, Hirsch P. A tabu search based solution method to optimize round timber transport. Proceedings of the MIC2005: The sixth Metaheuristic International Conference, Vienna, Austria, August 22-26, 2005.

[13] Ho SC, Haugland D. A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. Computers & Operations Research 2004; 31: 1947-1964.

[14] Murphy G. Reducing trucks on the road through optimal route scheduling and shared log transport service. Southern Journal of Applied Forestry 2003; 27(3): 198-205.

[15] Ropke S, Pisinger D. An adaptive large neighborhood search heuristics for the pickup and delivery problem with time windows. Transportation Science 2006; 40(4): 455-472.

[16] Ropke S, Pisinger D. A unified heuristic for a large class of vehicle routing problems with backhaul. European Journal of Operational Research 2006; 171: 750-775.

[17] Palmgren M. Optimal truck scheduling - mathematical modeling and solution by the column generation principle. PhD Thesis, Dissertation No. 967, Linköping University 2005.

[18] Palmgren M, Rönnqvist M, Värbrand P. A solution approach for log truck scheduling based on composite pricing and branch and bound. International Transactions of Operations Research 2003; 10(3): 433-448.

[19] Palmgren M, Rönnqvist M, Värbrand P. A near-exact method for solving the log-truck scheduling problem. International Transactions of Operations Research 2004; 11(4): 447-464.

[20] Rönnqvist, M. Optimization in forestry. Mathematical Programming Ser. B 2003; 97: 267-284.

[21] Rönnqvist M, Ryan D. Solving Truck Despatch Problem in Real Time. In: Proceedings of the 31th Annual Conference of the Operational Research Society of New Zealand, August 31-September 1, Wellington, New Zealand, 1995. p. 165-172.

[22] Rönnqvist M, Sahlin H, Carlsson D. Operative planning and dispatching of forestry transportation, Report LiTH-MAT-R-1998-18, Linköping University, Sweden, 1998.