International Conference
on Industrial Engineering and Systems Management
IESM 2009
May 13-15
MONTRÉAL - CANADA

A Heuristic to Solve the Weekly Log-Truck Scheduling Problem

Nizar El hachemi [a], Michel Gendreau [a], Louis-Martin Rousseau [a]

[a] *Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)*
C.P. 6128, succursale centre-ville, Montreal, Canada H3C 3J7

Abstract

We present in this article the log-truck scheduling problem, which combine routing and scheduling of trucks with some specific constraints related to the canadian forestry context. This problem includes aspects such as pick-up and delivery, multiple products, inventory stock, multiple supply points and multiple demand points. We developped a decomposed approach to solve the weekly problem, in two phases. At the first phase we use a tabu search algorithm to solve an integer problem in order to determine the destinations of full truckloads from forest areas to woodmills. At the second phase, we make use of a standard local search algorithm to schedule the daily transportation of logs. This approach has been implemented using COMET 0.07 that use the concept of constraint-based local search. We tested our method on a set of industrial cases from forest companies in canada.

Key words: Forestry, transportation, routing, scheduling, local search , constraint-based local search

1 Introduction

Scheduling problems in the forest industry have received significant attention in the recent years, essentially for economic and environmental reasons. Actually, the schedule of trucks is carried out manually by a specialist planner of the company.

The problem consist of a supply of differents products at many foret areas and a demand at differents woodmills. The volumes of wood are expressed in unit of truckload at both supply and demand points. In our case, there is no time windows at both forest areas and woodmills, trucks and loaders must be synchronised as much as possible to avoid waiting time. Demand at woodmills is given on a daily basis, whereas routes and schedules of trucks are to be found on a weekly basis. A particular constraint is present in our problem; each truck visits only one forest area and one mill in any given trip; it thus operates in a truck-load setting. We assume that at each supply point there is a single log-loader that ensures the loading of trucks (idem for demand points). There is also a constraint of stock at each woodmill (per product) which implies an integration between days.

The log-truck scheduling problem (LTSP) is closely related to some routing problems encountered in other industries, in particular, so-called "pick-up and delivery problems" with time windows. In general,

Email addresses: nizar@crt.umontreal.ca (Nizar El hachemi), michelg@crt.umontreal.ca (Michel Gendreau), louism@crt.umontreal.ca (Louis-Martin Rousseau).

LTSP is more complex than the classical PDPTW, difference comes from fact that at the LTSP we synchronise trucks and log-loaders. For general surveys of VRP and PDPTW, we refer the reader to [Cordeau et al. 2002] and [Gendreau et al. 2002].

Several models and methods have been developped in the litterature to solve the LTSP.
In chile [Weintraub et al. 1996] proposed a heuristic-based model (ASICAM) which worked succesfully since 1990 by producing a daily plan for trucks. [Linnainmaa et al. 1995] developped EPO, that deals with all stages from strategic to operative planning, its main output is a weekly schedule for each truck. [Palmgren et al. 2003] proposed a column-based routing model, solved by using branch and price. The column generation is based on heuristic enumeration that uses the result of a LP-based flow problem. [Flisberg et al. 2008] proposed a two-phase solution approach which transforms the logging trucks problem into a standard vehicle routing problem with time windows. the first phase determine the destination of flow form supply points to demand points, while the second phase combine transport nodes to routes. [Andersson et al. 2008] have used the same approach for planning the routes of trucks, but the main difference consists on the dispatching part procedure which is based on the previous work of [Rönnqvist et al. 1998]. [Gronalt and Hirsch 2007] applied a tabu search algorithm to solve the LTSP, a fixed set of destination transports is assumed. The same assumption is used by [El Hachemi et al. 2008] to develop a hybrid CP / IP method, where the IP model generates optimal routes in term of deadhead, while CP deals with the scheduling part. The principale contribution of this paper is to present an extension to the weekly time horizon where inventory stock context have taken into consideration.

2 Solution Approach

We propose in this paper, a two-phase hybrid method: in the first one, we solve an IP formulation of a tacticle problem, at this step we handled demand constraints (per product) at each woodmill and every day of the week, and stock constraints at each demand point (per product and per day), which imply integration between days. We introduce a capacity constraint at each supply point, in order to ensure loading feasibility for each day (there is only one log-loader at each forest area). Finally, we add a constraint for each forest area that ensure a minimal working time related to the associated log-loader. At this level, restrictions are on time availability of trucks.

Seven daily scheduling problems (ranged from Monday to Sunday) result from solving the first phase. For each daily problem, there is a fixed set of transportation requests to perform. The daily problems are solved sequentially, at this level, only one integration between two succesive days must be respected, each truck finishes its day at a woodmill and starts it on the same woodmill the next day.

We developped two local search algorithms enhanced with a tabu component for the both phases. We choose to move to the first best-improvement neighbor for the two phases (we select the first move that improves the current solution), and when the neighborhood does not improve the current solution we accept to move to a neighbor that deteriorate the current solution. More details about our algorithms will be evoked in the coming sections.

3 The Tactical Model

We present at this section our planning model and its associated local search algorithm.

3.1 Parameters

F	:	The set of forest areas.
W	:	The set of woodmills.
P	:	The set of forest produits.
J	:	The set of days of the week.
$Slimit$	:	The limit of full truckloads which can be stocked per product, per woodmill and per day.
U	:	The upper bound of full truckloads which can be transported from any supply point per day.
L	:	The lower bound of full truckloads that must be transported from any supply point per day if it is opened.
D^j_{wp}	:	The demand of product p at woodmill w, the day j.
C^*	:	The fixed cost of opening any forest area for one day.
c_{fw}	:	The travelling cost betweem a forest area f and a woodmill w.

3.2 Variables of the Model

$open^j_f$	:	A binary variable being true if the forest area f is opened the day j.
$trip^j_{fwp}$	:	An integer variable representing the number of full truckloads of product p from supply point f to woodmill w the day j.
$stock^j_{wp}$	:	An integer variable representing the number of full truckloads of product p stocked at the woodmill w the day j.

3.3 Constraints and Objective

$$\text{Minimize} \sum_{j\in J}\sum_{f\in F} C^* open^j_f + \sum_{j\in J}\sum_{f\in F}\sum_{w\in W} c_{fw} trip^j_{fwp}$$

$$\begin{aligned}
\text{subject to} & & \\
stock^j_{wp} &\leq Slimit,\ \forall w \in W, \forall p \in P, \forall j \in J && (1)\\
stock^{j-1}_{wp} + \sum_{f\in F} trip^j_{fwp} &= D^j_{wp} + stock^j_{wp},\ \forall w \in W, \forall p \in P, \forall j \in J && (2)\\
\sum_{w\in W}\sum_{p\in P} trip^j_{fwp} &\leq open^j_f U,\ \forall f \in F, \forall j \in J && (3)\\
\sum_{w\in W}\sum_{p\in P} trip^j_{fwp} &\geq open^j_f L,\ \forall f \in F, \forall j \in J && (4)
\end{aligned}$$

At this step, the objective function is to minimize the cost of opened sites and the full truckloads travel cost. We decide to choose $C^* >> c_{fw}$. This choice is based on the fact that the optimizer will seek to minimize the number of sites in operation each day. This will involve a global control over log-loaders waiting times each day, since no costs are attributed to inactive loader during the whole day.

Constraint (1) ensures that the daily stock of any product at any woodmill respects the maximum limit fixed. Constraint (2) specifies that demand is satisfied over all the week. Constraint (3) expresses the fact that each loader has a finite capacity to load logs (a loader cannot exceed a number of loading per day). Finally, constraint (4) implies that if a loader is operating during one day, it must work at least for certain time.

3.4 The Local Search Algorithm Adopted to the Tactical Problem

We have developped a local improvement algorithm with a tabu component to solve the tactical model. This algorithm is divised on two parts executed sequentially: at the first part, we generate an initiale solution by supposing that all forest areas are opened all the week, and applying a min cost flow algorithm. This is possible because, once the binary variables $open_f^j$ are fixed, the tactical problem became a simple min cost flow problem.

We denote by s a solution of the tactical model, we choose to represent s only by its binary variables $open_f^j$, $s =< open_1^1, ..., open_{|F|}^{|J|} >$. The neighborhood used at the first part is simple: it consist of changing the status of any opened forest area at any day to the closed status. $N_1(s) = \left\{ flip(s, open_f^j) \mid open_f^j = true \right\}$. We have two one-dimentional array tabu list, one associated to the forest areas and the other associated to the days. If the binary variable $open_f^j$ is related to a move $s^* \rightarrow s$, we can make this move tabu by imposing that the forest area f, and the day j to be prohibited for a few iterations.

After the completion of the first part of the adopted tactical algorithm (see figure 1), its second part is executed with the goal of minimizing the loaded travel cost. At this step the number of opened forest areas during the week remain constant, this is true since the construction of neighborhood N_2 associated to it respects this property. The second part keeps the same spirit as in the first one with three basic differences. The first one is that we accept to move to a solution that deteriorate the objective function instead of perturbing the current solution as in the first part. The second one consist of using a min cost flow algorithm instead of a compatible flow algorithm. Finally, the objective function at the second part represents only the loaded travel cost (see figure 2).

The neighborhood N_2 of the second part consist of selecting any two differents forest areas f_1 and f_2, and a day j such that $open_{f_1}^j$ and $open_{f_2}^j$ have a differents values, and flip their values. $N_2(s) = \left\{ flip(s, open_{f_1}^j, open_{f_2}^j) \mid open_{f_1}^j = true, open_{f_2}^j = false \right\}$. We have used the same approach to make a move tabu as in the first part of the adopted tactical algorithm. Concerning the deterioration percentage ϵ that we tolorate to accept a move, we decide to change it dynamically during the second part.

Finally, at the output of this step, seven input data associated to each day of the week representing logs to transport from any forest area to any woodmill will be communicated to different daily log-truck scheduling problem of the week. In the next section, we will present the daily log-truck scheduling problem with a new model adapted to the spirit of constraint-based local search.

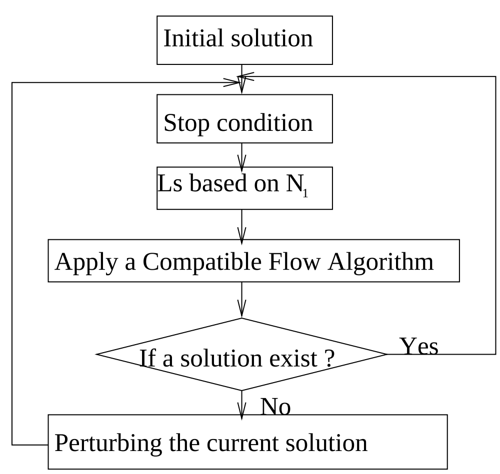

Fig. 1. First part of the tactical problem algorithm

4 The Daily Log-Truck Scheduling Problem

At this section, we will descript our log-truck scheduling model and the local search algoritm developped to solve it.

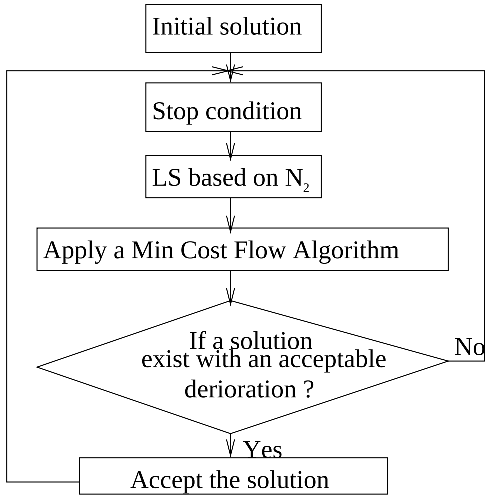

Fig. 2. First part of the tactical problem algorithm

4.1 Model Parameters

$D[m,f]$	:	The distance between a mill m and a forest area f.
nbR	:	The number of the real transportation requests.
nbV	:	The number of vehicles.
R	:	The set of real transportation requests $[1..nbR]$.
I	:	R added with departures from initiale locations $[(1-nbV)..nbR]$.
O	:	R added with empty returns to finale locations $[1..(nbR+nbV)]$.
T	:	The deadline to tranport all requests.
c_t	:	The cost of waiting time of a truck per unit of time.
c_l	:	The cost of waiting time of a log-loader per unit of time.
c_d	:	The cost of an empty driving per unit of time.
d_p	:	The duration of a pick-up activity at a forest area.
d_d	:	The duration of a delivery activity at a woodmill.

4.2 Model Variables and Invariants

$Variables$	:	
$Start_{P_r}$	:	The beginning time of the pick-up activity related to the request r.
$Start_{D_r}$	:	The beginning time of the delivery activity related to the request r.
$Prec_r$	:	The request preceding the request r on their associated truck route.
$Invariants$	:	
End_{P_r}	:	The ending time of the pick-up activity related to the request r.
End_{D_r}	:	The ending time of the delivery activity related to the request r.
$Wait_{M_r}$	:	The waiting time of a truck at a woodmill related to the resquest r.
$Wait_{F_r}$	:	The waiting time of a truck at a forest area related to the resquest r.
$Start_{F_f}$	:	The starting time of work of the log-loader associated to the forest area f.
End_{F_f}	:	The ending time of work of the log-loader associated to the forest area f.

Invariants, or one-way constraints (→), are at the core of the LS module COMET architecture. They provide declarative specifications of incremental algorithms that are fundamental in obtaining high performance for local search. More precisely, invariants specify what to maintain incrementally, not how to do so, which is the role of the COMET implementation (see [Van hentenryck and Michel 2001] and [Van hentenryck and Michel 2006]).

4.3 Model Formulation of the Log-Truck Scheduling Problem

We descript our log-truck scheduling model with the objective function to minimize the cost of unproductive activities.

$$\text{Min} \sum_{r \in R} c_d D[M_{Prec_r}, F_r] + \sum_{r \in R} c_t (Wait_{M_r} + Wait_{F_r}) + \sum_{f \in F} c_l (End_{F_f} - Start_{F_f})$$

$$\begin{aligned}
&\text{subject to} && \\
Wait_{M_r} &\rightarrow Start_{D_r} - D[M_r, F_r] - End_{P_r},\ \forall r \in R && (5)\\
Wait_{F_r} &\rightarrow Start_{P_r} - D[M_{Prec_r}, F_r] - End_{D_{Prec_r}},\ \forall r \in R && (6)\\
End_{P_r} &\rightarrow Start_{P_r} + d_p,\ \forall r \in R && (7)\\
End_{D_r} &\rightarrow Start_{D_r} + d_d,\ \forall r \in R && (8)\\
\left|Start_{P_{r_1}} - Start_{P_{r_2}}\right| &\geq d_p,\ \forall r_1, r_2 \in R : F_{r_1} = F_{r_2} && (9)\\
\left|Start_{D_{r_1}} - Start_{D_{r_2}}\right| &\geq d_d,\ \forall r_1, r_2 \in R : M_{r_1} = M_{r_2} && (10)\\
End_{D_r} &\leq T,\ \forall r \in R && (11)\\
Start_{F_f} &\rightarrow \min(Start_{P_r}),\ \forall r \in R : F_r = f && (12)\\
End_{F_f} &\rightarrow \max(End_{P_r}),\ \forall r \in R : F_r = f && (13)\\
&\textsc{alldifferent}(\text{Prec}) && (14)\\
Prec_r &\neq r,\ \forall r \in R && (15)\\
Wait_{M_r} &\geq 0,\ \forall r \in R && (16)\\
Wait_{F_r} &\geq 0,\ \forall r \in R && (17)
\end{aligned}$$

Constraints (16) and (17) express the fact that a truck waiting time at woodmills and forest areas are non negative. Constraints (5) and (6) express the invariants related to waiting time of trucks at woodmills and forest areas respectively. Combining (16) and (5) ensures that a truck transporting request r has enough time to do it. Combining (17) and (6) ensures that a truck that carries out two successive requests (including the dummy departure) has enough time to do so. Constraints (7) and (8) represent the invariants related to the end of time of the pick-up and the delivery activity respectively. Constraints (9) and (10) ensure that the log-loaders at forest areas and woodmills have enough time to load and unload any request. Constraints (12) and (13) express the invariants computing the starting time of the first pick-up activity and the end of time of the last pick-up activity associated to a forest area. Constraint (11) specifies that we respect the deadline to transport all the requests. Constraint (14) specifies that all predecessors are given a different value (this combined with the fact that a predecessor variable can only have one value, is equivalent to flow conservation constraint). Finally, constraint (15) expresses that a predecessor of a request cannot be the request itself.

4.4 Solving the LTSP

Our problem has two integrated difficults aspects (routing and scheduling), we developped a heuritic algorithm separating the routing and the scheduling on two sequential stages. This heuritic is based on a local search operating on the trucks routes, and a greedy schedule respecting the routes of the previous stage. A tabu component have taken into consideration to enhance the local search algorithm. Some new parameters are defined as follow to facilitate the comprehension of the adopted neighborhood structure.

Veh : The set of trucks.

l_{ik} : The k^{th} shipment belonging to the i^{th} truck route.

l_i : The most costly log in term of truck waiting time belonging to the i^{th} truck route.

We define a neighborhood based on permuting two logs belonging to two different trucks routes. When the selection of two different trucks is realised, we fixed one route and choose its most costly shipment, in term of truck waiting time, and permute it with any other shipment belonging to the other route. We note the neighborhood structure by N. $N(s) = \{swap(l_i, l_{jk})\ \forall(i, j \neq i) \in Veh^2,\ \forall k\}$. Of course, the selection of trucks and logs must not be tabu.

In practice, the initiale assignment of values to the vector $Prec$ satisfaies the two constraints (14) and (15).

$$Prec_r = r - nbV,\ \forall r \in O$$

Combining this initiale assignment and the struture of the neighborhood N, we can remove this two constraints (14) and (15) from our LTSP model, as we are sure that this last constraints will be satisfied all time during our local search algorithm.

The best-improvement heuritic requires a complete scan of the neighborhood N, this is expensive in terms of comptutational time. We choose the first-improvement heuristic, which simply selects the first move that improves the current solution. In our context, since we limited the computational time (number of iterations), the first-improvement heuristic provides better solutions than the best-improvement heuristic. This is explained by the fact that the first-improvement heuristic visits more regions of the search space.

We have an one-dimentional array list, listing tabu trucks, and a two dimentional matrix, storing inverse swap of logs. Thus, after each move, we have to ensure that the trucks and the inverse swap of logs involved in the move will be tabu for a few iterations. After same iterations, if we did not improve the current solution, we accept to move to a new solution that deteriorate the current solution. We impose that the deterioration percentage must not exceed certain limit (this limit can be changed dynamically during the algorithm).

The intensification component is based on a simple idea of restarting the search from the best solution found so far when no improvement took place in the last n iterations. We apply a greedy algorithm to schedule all activities of loading and unloading. See figure 3 for more details about our local search algorithm.

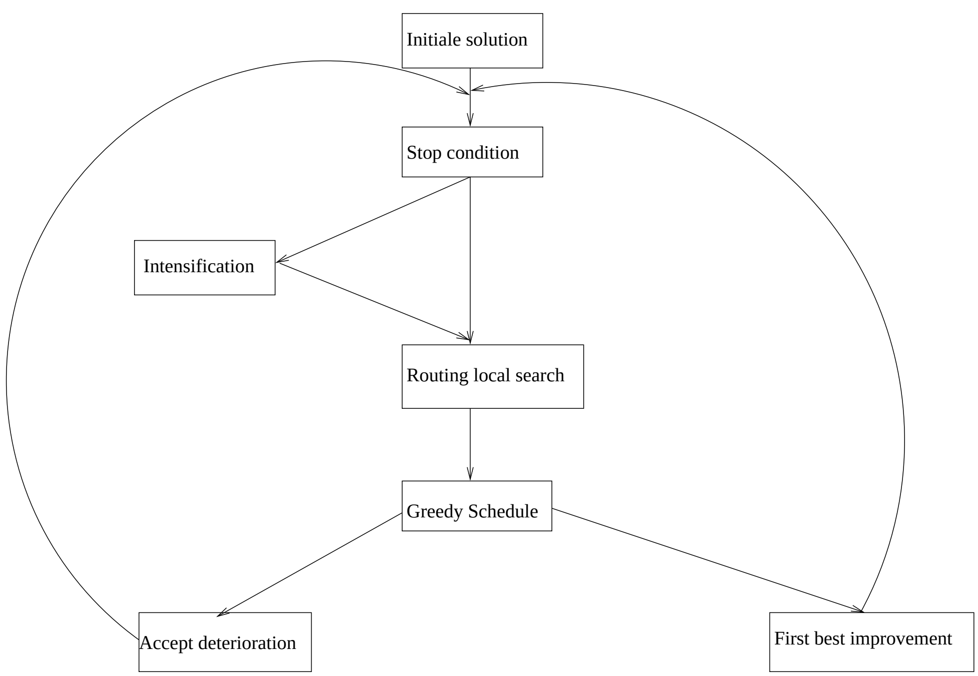

Fig. 3. LTSP local search

5 Experimental Results

We tested our algorithmic approach on an industrial case study with two different instances. This case is provided by the Forest Engineering Research Institute of Canada (FERIC). It involves six forest areas and five woodmills, its related average cycle time to transport a shipment is about 4 hours. The two instances come from a larger timber company and have respectively 13 and 14 trucks, and about 400 shipments. For the both instances, loading and unloading take about 20 minutes, that's why we choose to discretize the data in 20 minutes. We used the following costs: 60$ per hour for the truck waiting time, 70$ per hour for the truck travel, and 100$ per hour for the log-loader waiting time . We ran each scenario five times and report the result of total cost at (Table 1) that shows the best solution, the worst solution, the average solution and the standard deviation from it. In the same spirit we report results concerning the loaded travel cost, the empty driven cost and finally, the waiting time cost.

In practice, the timber company performs a manual routing and scheduling, and complain that they suffer from large waiting times. We outperform the manual strategy, this is explained by the adopted strategy of the first phase that minimize the number of opened forest areas during the week, which is not the case of the manual strategy, which uses in general much more opened forest areas during the week than it is necessary. Unfortunatelly, we don't have the transportation part cost of the manual strategy performed by the company.Looking at Table 1, we see that the transportation cost is much higher than the waiting time cost. Thus a small reduction of such a large cost would represent a considerable gain.

Globaly, our algorithm is stable (standard deviation from the average solution is less than 1% of it for the both instances), but what is less stable is the waiting time part cost (see Table 1), where STD represents about 4% of the average solution related to the first instance. We believe that a diversification of neighborhoods (combining a various neighborhoods) will improve the solution quality and make it more stable.

Table 1
Real Data Results.

Instances	Total cost ($)	Loaded travel cost ($)	Empty driven cost ($)	Wating time cost ($)
Best solution				
1	65446	27510	29353	8373
2	63260	26716	28816	7673
Worst solution				
1	67366	28000	30193	9440
2	64780	27743	29820	8006
Average solution				
1	66420	27773	29786	8856
2	64256	27143	29296	7813
(Standard deviation / Average solution)				
1	0,96%	0,62%	1,17%	4,1%
2	0,81%	1,43%	1,2%	1,45%

6 Conclusion

We have presented the weekly log-truck scheduling problem with an objective function to minimize transportation and unproductive time cost. To adress this problem, we decomposed it into two phases. The first phase determines the destination of full truckloads (from forest areas to woodmills) by minimizing the transportation and opening forest area cost. At the output of this phase seven daily problem are generated covering all the week and solved sequentially. The main of the second phase is to schedule the daily transportation of logs by optimizing the cost of unproductive time.

Other research directions involve solving the second phase using an iterated local search algorithm that use a greedy schedule to evaluate solutions in various neighborhoods. Other research tracks consist on adding a weekly supply constraint at each forest area , this will impose a fundamental change in the structure of our heuristic adapted to the first phase.

References

[Andersson et al. 2008] G. Andersson, P. Flisberg, B. Lidén, and M. Rönnqvist (2008). RuttOpt- a decision support system for routing of logging trucks. *Candian Journal of Forest Research, 38* 1784–1796.

[Cordeau et al. 2002] J. F. Cordeau, G. Desaulniers, J. Desrosiers, M. M. Solomon, and F. Soumis (2002). VRP with time windows. *In Proceeding. Toth and D. Vigo (eds) : The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications, vol 9, Philadelphia, PA* 157–193.

[El Hachemi et al. 2008] N. El Hachemi, M. Gendreau, and L. M. Rousseau. Solving a Log-Truck Scheduling Problem with Constraint Programming (2008). *In Proceeding of the Fifth International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems* 293–297.

[Flisberg et al. 2008] P. Flisberg, B. Lidén, and M. Rönnqvist. A hybrid method based on linear programming and and tabu search for routing of logging trucks (2008). *Computers and Operations Research, 36(4)* 1122–1144.

[Gendreau et al. 2002] M. Gendreau, G. Laporte, and J. Y. Potvin (2002). Metaheuristics for the Capacitated VRP. *In Proceeding. Toth and D. Vigo (eds): The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications, vol 9, Philadelphia, PA* 129–154.

[Gronalt and Hirsch 2007] M. Gronalt, and P. Hirsch (2007). Log-Truck Scheduling with Tabu Search Strategy. *Metaheuristics, Vol 39* 65–88.

[Linnainmaa et al. 1995] S. Linnainmaa, J. Savalo, and O. Jokinen (1995). EPO: A knowledge based system for wood procurement management. *Paper presented at the 7th Annual Conference on Artifical Intelligence, Montreal*.

[Palmgren et al. 2003] M. Palmgren, M. Rönnqvist, and P. Värbrand (2003). A solution approach for log truck scheduling based on composite pricing and branch and bound. *International Transactions in Operational Research, 10* 433–447.

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

[Van hentenryck and Michel 2001] P. Van hentenryck, L. Michel (2001). Constraint-Based Local Search. MIT Press.

[Van hentenryck and Michel 2006] P. Van hentenryck, L. Michel (2006). Differentiable Invariants. *In Proceeding of the Twelfth International Conference on Principles and Practice of Constraint Programming (CP06)* 685–689.

[Weintraub et al. 1996] A. Weintraub, R. Epstein, R. Morales, J. Seron, and P. Traverso (1996). A truck scheduling system improves efficiency in the forest industries. *Interfaces (26),* 1–12.