

A Heuristic to Solve the Synchronized Log-Truck Scheduling Problem

Nizar El Hachemi, Michel Gendreau and Louis-Martin Rousseau

Interuniversity Research Centre on Enterprise Networks,
Logistics and Transportation (CIRRELT)
C.P. 6128, succursale centre-ville, Montreal, Canada H3C 3J7
{nizar,michelg,louism}@crt.umontreal.ca

Abstract

We present a synchronized routing and scheduling problem that arises in the forest industry, as a variation of the log-truck scheduling problem. It combines routing and scheduling of trucks with 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 developed a decomposition approach to solve the weekly problem in two phases. In the first phase we use a MIP solver to solve a tactical model that determines the destinations of full truckloads from forest areas to woodmills. In the second phase, we make use of two different methods to route and schedule the daily transportation of logs : the first one consists in using a constraint-based local search approach while the second one is a hybrid approach involving a constraint programming based model and a constraint-based local search model. These approaches have been implemented using COMET2.0. The method, was tested on two industrial cases from forest companies in Canada.

keywords: Synchronized routing, scheduling, forestry, local search, constraint-based local search and constraint programming

1 Introduction

Forestry represents a major economic sector in Canada and account for a large proportion of its exports. Planning problems in forestry cover a wide scope of activities ranging from planting and harvesting to road building and transportation. Furthermore, in many of these activities, it is critical to pay attention to important environmental issues, such as green house gas emissions, as well as to company-specific goals, operations rules, and government restrictions.

In Quebec, transportation represents more than 30% of the cost of provisioning for wood transformation mills, i.e., approximately \$15 per cubic meter of roundwood. The average distance between forest areas where wood is collected and mills to which this wood is transported is around 150 km, and about 50% of the fuel required per cubic meter of collected wood is consumed by the forest trucks traveling, half of the time empty, between forest areas and mills.

Transport activities between forest areas and mills should therefore be organized as effectively as possible. In recent years significant attention has been devoted to transportation-related scheduling problems, mainly for economic and environmental reasons, in most Canadian forest companies, truck schedules are derived manually by an expert planner.

The main contributions of this paper are to synchronize trucks and log-loaders and to extend the log-truck scheduling problem (LTSP) to the weekly horizon, where inventories at wood mills are taken into consideration in order to allow woodmills to work in a just in time mode. Furthermore, our approach is generally applicable to problems with a need to synchronize routing and scheduling.

The paper is organized as follows. Sections 2, 3, 4 and 5 present respectively the previous work, the problem description and a brief description of the solution approach, the tactical model and the daily Synchronized LTSP. The experimental setting is described in Section 6, where computational results are also reported. Section 7 concludes the paper.

2 Previous work

The LTSP is closely related to some routing problems encountered in other industries, in particular, so-called “pick-up and delivery problems”. For general surveys of the vehicle routing problem (VRP) and pick-up and delivery with time windows (PDPTW), we refer the reader to Cordeau *et al.* [6], Gendreau *et al.* [13], Savelsbergh and Sol [33], Dumas *et al.* [7] and Toth and Vigo [35]. This paper considers a pick-up and delivery problem in which for each request exactly one load of wood has to be transported from its pick-up location (forest area) to its delivery location (woodmill). A truck visits only one forest area and one mill on any given trip, i.e., requests are served individually by trucks. After unloading at a mill from its previous trip, a truck is usually sent back empty to its next forest destination.

Several models and methods have been developed in the literature to solve the LTSP. Among these, the heuristic-based approach of [38] (ASICAM) has been used successfully since 1990 to produce daily plans for trucks in Chile. The application of ASICAM in Chile has led to a 32% reduction in truck fleet size, a 13% reduction in average working hours and operational costs, and a 31% increase in productive hours for one company. Linnainmaa *et al.* [21] propose a knowledge-based system called EPO that deals with all stages from strategic to operational planning. The input data is collected on-line directly from the forest areas and the main output is a weekly schedule for each truck. One main goal of EPO is the minimization of truck driving as whole. Palmgren *et al.* [26] have proposed more recently a pseudo column generation based model, which is solved using a branch-and-price procedure, where each column represents

one feasible route for one truck. An initial set of routes is generated at the beginning. The subproblem, which is a constrained shortest path problem, is then solved by applying a k-shortest path algorithm. The columns whose reduced costs are negative are communicated to the master problem. Flisberg *et al.* [12] and Andersson *et al.* [1] proposed a two-phase solution approach, in which the LTSP is transformed into a standard vehicle routing problem with time windows. In the first phase, flows are determined from supply points to destinations, thus yielding a number of transportation “tasks” that are combined into routes in the second phase. The dispatching procedure that continuously updates the trucks routes during the day is based on the previous work of [30] and [29]. Gronalt and Hirsch [14] applied a tabu search algorithm to solve a restricted variant of the LTSP in which the number of trips between each forest area and each mill is given. The same assumption is used by [8] who develop a hybrid method combining Integer Programming(IP) and Constraint Programming(CP): the IP model generates optimal routes in term of deadheading, while CP deals with the scheduling part. For a more detailed description of optimization problems in the forest sector, we refer the reader to [28].

With respect to numerical experiments, [24] presents a case study with on average 9 trucks and 35 transport tasks per day, while [25] have solved two case studies for Sweden: one with 6 trucks and 39 transport tasks, and one with 28 trucks and approximately 85 transport tasks. Gronalt and Hirsch [14] solved random problems with 30 transport tasks and 10 trucks, [8] deal with daily problems having up to 18 trucks and 70 transport tasks. Finally, [12] and [1] have solved substantially larger instances ranging from 188 transport tasks to about 2.500 full truck loads with 15 to 110 trucks.

Synchronizing vehicles with others tasks arises in many real-world applications. The synchronization constraints occur in the well known dial-a-ride problem (see [32]), in which some disabled persons require assistance in order to prepare for transportation. This assistance can vary from dressing to providing a wheelchair and is not usually required at all time or by all customers. More details concerning the homecare staff scheduling are given by [9] and [10]. There are many other applications in which vehicles must be synchronized, like the urban mass transit systems, which combines vehicle and crew scheduling problems and where drivers are allowed to change bus in so called relief points. As described in [15] and [11], the synchronized arrival of bus to the relief points is implicitly guaranteed, since it is defined by a timetable. In the airline industry, synchronization constraints have been introduced by [19]. In fact, the classical fleet assignment problem studied by [16] states that, given a fixed flight timetable, there is a need to decide which aircraft to use for each flight on a daily basis. However, [19] studied the case where the timetable is not fixed and departure times of flights are given on a daily basis

for an entire week, such that each flight has a time window on either its departure or its arrival. To obtain a more robust schedule, synchronization constraints are introduced to guarantee that every flight will have the same departure time on each day. In the forest industry, harvesting operations are performed by harvesters and forwarders. The coordination between them specifies that forwarding can only be done once harvesting has been performed (see [20]). In the same context, and related to the LTSP, there is a need to synchronize trucks that do not have an onboard crane with forest loaders. For more details about the importance of synchronization constraints in the real-world situations, we refer the reader to [4].

3 Problem Description and General Solution Approach

It must be noted that the problem as considered in this paper differs in several ways from LTSPs addressed by other authors. These differences stem from specific characteristics of Canadian forestry operations. In Canada, cut areas and log volumes are generally quite large. It is therefore customary when dealing with higher value products, such as hard wood, to assemble full truckloads prior to transportation by merging similar logs that are going to the same destination. While quantities are generally expressed in cubic meters, these can be easily converted into truckloads. In several variants of the LTSP, each truck must begin and end its route at its given home base, which often corresponds to the truck operator’s home or yard. In our application, the bases of all trucks are mills and we are allowed to reallocate trucks among mills to obtain more efficient solutions. Driver changes are normally performed at mill when a route exceeds the maximum allowed driving time for a driver. In this paper, issues related to driving time, rests or changeovers are not taken into account. Demands at woodmills are given on a daily basis, whereas routes and schedules of trucks are determined on a weekly basis. Since the stock of products at woodmills are constrained, this requires integrating transportation schedules over several days.

We assume that at each mill and each forest location, there is a single log loader that ensures the loading and unloading of all trucks. When a truck arrives at a location, if the loader is busy, the truck has to wait until the loader becomes available. These waiting times can severely delay trucks and thus increase the cost of transportation; they should therefore be avoided as much as possible. In some variants of the LTSP studied in the literature, loaders work a fixed shift during the day, and it is assumed that they support harvesting between truck loading operations. In our application, loaders cannot effectively support harvesting operations between successive trucks as forest sites are too large, it is thus critical to maximize their usage.

The solution approach that we have implemented is based on a decomposition of the overall

weekly problem into a "tactical problem", in which the destination of each full truckload of wood is determined, and a series of seven "daily problems" in which we determine deadhead trips and schedule trucks and forest loaders.

The proposed approach takes into account the cost of loaded travel in the tactical phase; once loaded trips have been determined, empty driven distance and waiting times in forest areas and at woodmills are minimized, which leads to a reduction of green house gas emissions, since these are directly proportional to fuel consumption.

4 The Tactical Problem

The tactical problem is addressed using a MIP model. As mentioned before, the purpose of the tactical model is to determine to which woodmill each load should be transported. A key element with respect to these decision is the fact that different wood products have to be shipped. Each product corresponds to given species, usage, quality and dimension, since woodmills often require logs of specific length and diameters to produce given final products. We now introduce the various components of our MIP model.

Parameters

- F : The set of forest areas.
- W : The set of woodmills.
- P : The set of forest products.
- J : The set of days of the week, $J = \{1, 2, 3, 4, 5, 6, 7\}$.
- S_p^{max} : The daily maximum number of full truckloads of product p that can be stocked at a woodmill.
- S_{wp}^i : The initial stock of product p stocked at woodmill w .
- U : The daily upper bound on the number of full truckloads that can be transported from any supply point.
- L : The daily lower bound on the number of full truckloads that must be transported from any supply point if it is opened.
- D_{wp}^j : The demand of product p at woodmill w on day j .
- nbL : The available forest loaders.
- c_{fw} : The travelling cost between forest area f and woodmill w .

Variables of the Model

- $open_f^j$: A binary variable equal to 1 if forest area f is opened on day j .
 $trip_{fwp}^j$: An integer variable representing the number of full truckloads of product p from supply point f to woodmill w on day j .
 $stock_{wp}^j$: An integer variable representing the number of full truckloads of product p stocked at woodmill w on day j , $j \in \{0, 1, 2, 3, 4, 5, 6, 7\}$.

Constraints and Objective

$$\text{Minimize } \sum_{j \in J} \sum_{f \in F} \sum_{w \in W} \sum_{p \in P} c_{fw} trip_{fwp}^j$$

subject to

$$stock_{wp}^j \leq S_p^{max}, \forall w \in W, \forall p \in P, \forall j \in J \quad (1)$$

$$stock_{wp}^0 = S_{wp}^i, \forall w \in W, \forall p \in P \quad (2)$$

$$stock_{wp}^{j-1} + \sum_{f \in F} trip_{fwp}^j = D_{wp}^j + stock_{wp}^j, \forall w \in W, \forall p \in P, \forall j \in J \quad (3)$$

$$\sum_{w \in W} \sum_{p \in P} trip_{fwp}^j \leq U \cdot open_f^j, \forall f \in F, \forall j \in J \quad (4)$$

$$\sum_{f \in F} open_f^j \leq nbL, \forall j \in J \quad (5)$$

$$\sum_{w \in W} \sum_{p \in P} trip_{fwp}^j \geq L \cdot open_f^j, \forall f \in F, \forall j \in J \quad (6)$$

In this phase, the objective function consists of minimizing the loaded travel cost subject to the following constraints. Constraint (1) ensures that the daily stock of any product at any woodmill respects the maximum limit fixed. Constraint (2) sets the initial stock of each product at each woodmill. Constraint (3) specifies that demand must be satisfied over the whole week. Constraint (4) expresses the fact that each loader has a finite capacity to load logs (a loader cannot exceed a number of loads per day). Constraint (5) limits the number of available forest loaders. Finally, constraint (6) specifies that if a loader is operating during a given day, it must serve a minimum number of trips.

We have used the MIP solver imbedded in COMET2.0 (SCIP) to solve the tactical model. For the two instances that we will describe later, we used only the default parameters of SCIP.

5 The Daily Synchronized Log-Truck Scheduling Problem

To address the daily LTSP and deal with synchronization constraints, we first developed a straightforward approach based on constraint-based local search (CBLS) and the COMET2.0 programming language. A second approach follows directly from the first one by hybridizing the CBLS model with a previous CP model for the LTSP (see [8]) in some specific situations.

CBLS consists in using constraints and objective function to describe and control the local search. The invariants (incremental variables), or one-way constraints provide declarative specifications of incremental algorithms, they have been introduced by [22] and [23]. More precisely, they specify what information should be maintained incrementally. They are useful for computing various neighborhoods efficiently and evaluating the impact of complex moves on differentiable objects (objective function). The project COMET was initiated in 2001 to explore the CBLS potentiality and how it could be supported in high-level programming languages (see [36]). Actually, COMET supports both modeling and search abstractions in the spirit of constraint programming and includes CP, CBLS and linear solvers.

CP is a versatile paradigm for modeling and solving various practical combinatorial optimization problems. Over the last years, constraint-based scheduling has become an important tool for modeling and solving scheduling problems (see [2], [3], [17] and [39]). In the context of the synchronized LTSP, it is a particularly attractive approach, because it can easily model the synchronization of trucks and log-loaders and the optimization of the waiting costs. CP represents loading and unloading times as domains variables and by propagating the synchronization constraints only when needed, it allows an efficient implementation of those constraints.

5.1 A Constraint-Based Local Search Model

To solve the CBLS model for the LTSP, we use an iterated local search algorithm (ILS). The ILS procedure integrates a vehicle routing algorithm based on various neighborhoods and a greedy algorithm to schedule the transportation of logs once routes are fixed. We now describe all components of our ILS algorithm. Thus we will present the CBLS model, the routing algorithm and its initial solution, and finally the greedy algorithm.

Parameters

- $D[m, f]$: The distance between a mill m and a forest area f .
- nbR : The number of real transportation requests.
- nbV : The number of vehicles.
- R : The set of real transportation requests $[1..nbR]$.
- F_r : The forest origin of request r .
- M_r : The woodmill destination of request r .
- I : R added with departures from initial locations $[(1 - nbV)..nbR]$.
- O : R added with empty returns to final locations $[1..(nbR + nbV)]$.
- T : The deadline to transport 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 deadhead cost per unit of time.
- D^+ : The duration of a pick-up activity at a forest area.
- D^- : The duration of a delivery activity at a woodmill.
- L^f : A log-loader assigned to forest area f defined as a unary resource.
- L^m : A log-loader assigned to woodmill m defined as a unary resource.
- W : The domain of waiting time variables $[0..T]$.

CBLS Model Variables

- $Start_{P_r}$: The beginning time of the pick-up activity related to request r .
- $Start_{D_r}$: The beginning time of the delivery activity related to request r .
- $Prec_r$: The request preceding the request r on their associated truck route.

It is noteworthy that for each day $j \in J$, the tactical and the daily models are connected explicitly by the following relationships : $nbR = \sum_{f \in F} \sum_{w \in W} \sum_{p \in P} trip_{fwp}^j$. Furthermore, in set R , we create $\sum_{p \in P} trip_{fwp}^j$ requests from forest area f to woodmill w for each pair (f, w) .

CBLS Model Invariants

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

CBLS Model and Formulation of the Synchronized LTSP

We describe our log-truck scheduling model with the objective function minimizing the cost of

unproductive activities.

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

subject to

$$Wait_{M_r} \leftarrow Start_{D_r} - D[M_r, F_r] - End_{P_r}, \forall r \in R \quad (7)$$

$$Wait_{F_r} \leftarrow Start_{P_r} - D[M_{Prec_r}, F_r] - End_{D_{Prec_r}}, \forall r \in R \quad (8)$$

$$End_{P_r} \leftarrow Start_{P_r} + D^+, \forall r \in R \quad (9)$$

$$End_{D_r} \leftarrow Start_{D_r} + D^-, \forall r \in R \quad (10)$$

$$|Start_{P_{r_1}} - Start_{P_{r_2}}| \geq D^+, \forall r_1, r_2 \in R : F_{r_1} = F_{r_2} \quad (11)$$

$$|Start_{D_{r_1}} - Start_{D_{r_2}}| \geq D^-, \forall r_1, r_2 \in R : M_{r_1} = M_{r_2} \quad (12)$$

$$End_{D_r} \leq T, \forall r \in R \quad (13)$$

$$Start_{F_f} \leftarrow \min(Start_{P_r}), \forall r \in R : F_r = f \quad (14)$$

$$End_{F_f} \leftarrow \max(End_{P_r}), \forall r \in R : F_r = f \quad (15)$$

$$\text{ALLDIFFERENT}(\text{Prec}) \quad (16)$$

$$Prec_r \neq r, \forall r \in R \quad (17)$$

$$Wait_{M_r} \geq 0, \forall r \in R \quad (18)$$

$$Wait_{F_r} \geq 0, \forall r \in R \quad (19)$$

Constraints (7) and (8) express the invariants (\leftarrow) related to the waiting time of trucks at woodmills and forest areas respectively. Constraints (9) and (10) represent the invariants related to the ending time of the pick-up and the delivery activity respectively. Constraints (11) and (12) specify that the log-loaders in forest areas and at woodmills have enough time to load and unload any request. Constraints (14) and (15) express the invariants computing the starting time of the first pick-up activity and the ending time of the last pick-up activity associated with any forest area. Constraint (13) ensures that we respect the deadline to transport all requests. Constraint (16) 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 a flow conservation constraint). Constraints (17) express that the predecessor of a request cannot be the request itself. Constraints (18) and (19) express the fact that waiting times are non negative. Finally, combining (18) and (7) ensures that a truck transporting request r has enough time to do so, while (19) and (8) ensures that there is enough time for a truck to carry two successive requests (including the dummy departure).

5.1.1 The Routing Operator

The routing algorithm proposed in this paper and incorporated in the ILS procedure is based on a set of neighborhoods, where each neighborhood represents a set of solutions obtainable from the current solution by applying some specified operations. The search space is the set of all ordered assignments of requests to trucks. The initial solution of ILS is $(Prec_r = r - nbV, \forall r \in O)$ that satisfies constraints (16) and (17).

We use some classical neighborhoods of the VRP literature (see [18]) and limit their size by using parameters; thus three different neighborhoods structures have been implemented.

1. i-opt neighborhood : The i-opt neighborhood (see [5] and [18]) was originally proposed for the traveling salesman problem (TSP). It consists in choosing a subpath (a sequence of some consecutive requests served by the same truck) of at most L_{path}^{iopt} (a parameter) loads, and moving it from its current position to another one in the same truck route, while preserving the same order.
2. 2-opt* neighborhood : This is a classical neighborhood for the VRP, which was introduced by [27]. It consists in selecting two different truck routes dividing each into two parts, and exchanging the second parts of the two routes.
3. cross-exchange neighborhood : Originally proposed by [34], it consists in exchanging two different subpaths of at most L^{cross} (a parameter) loads belonging to two different truck routes, while preserving the order of requests.

Among the neighborhoods considered, the last two deal with two different trucks routes. In order to reduce the computational effort, we make sure to select at least one of these routes among the 25% that have the largest truck waiting times.

5.1.2 A Greedy Scheduling Algorithm

The greedy algorithm performs the scheduling of activities (loading and unloading) taking into account the synchronization constraints. The algorithm is described as follows:

- Step 1 : Schedule all activities as soon as possible without considering the fact that loaders are unary resources, and put all loading activities in a list. Go to step 2.

- Step 2 : Select the earliest loading activity belonging to the list, and if there is any conflict involving the associated loader, correct it by pushing back conflicting activity. Propagate this correction along the route of the associated truck. Finally, remove this loading activity from the list.
- Step 3 : Select the unloading activity associated with the loading activity of step 2. If there is any conflict involving the associated loader, correct it as above. Go to step 4.
- Step 4 : If the list of loading activities is not empty, return to step 2.

5.1.3 The ILS algorithm

We adapted the ILS of [18] to our context, and we applied each neighborhood until it reaches a local minimum, such as illustrated in Figure 1. We use the objective as the evaluation function, and we move to the best solution related to the current neighborhood if it improves the current best solution. If none of the neighborhoods are able to improve the current best solution, then this solution is perturbed. Perturbing the current best solution consists in applying a cross exchange operation, which randomly chooses two subpaths from different routes and exchanges them. In order to generate a new initial solution, we select a number r randomly from the set $\{1, 2, 3, 4\}$, and perform consecutively r random cross exchange operations. During the process, neighborhoods are considered in the order presented in 5.1.1. Once ILS finds a better solution in a neighborhood, it returns to the i-opt neighborhood.

5.2 A Hybrid Approach

The decision variables of the CP model are based on the COMET2.0 CP solver. For each activity A , two finite domain variables are created, A^s and A^e , which are associated respectively with the beginning and the end of the activity.

5.2.1 CP Variables and Domains

- $P_r, \forall r \in O$: A pickup activity of duration D^+ if $r \in R$, and of duration 0 if $r \in (O \setminus R)$.
 $D_r, \forall r \in I$: A delivery activity of duration D^- if $r \in R$, and of duration 0 if $r \in (I \setminus R)$.
 $O_f, \forall f \in F$: An activity of duration equal to the opening time of forest area f .
 H : A dummy task activity of duration 0 that follows all other activities.
 $W_r^{M_r} \in W, \forall r \in R$: The waiting time of the truck transporting the request r at its related woodmill.
 $W_r^{F_r} \in W, \forall r \in R$: The waiting time of the truck transporting the request r at its related forest area.

5.2.2 CP Formulation of the Log-Truck Scheduling Problem

$$\text{Min } \sum_{r \in R} c_d D[M_{Prec_r}, F_r] + \sum_{r \in R} c_t (W_r^{F_r} + W_r^{M_r}) + \sum_{f \in F} c_l (O_f^e - O_f^s)$$

$$P_r \Leftrightarrow L^{F_r}, \forall r \in R \quad (20)$$

$$D_r \Leftrightarrow L^{M_r}, \forall r \in R \quad (21)$$

$$O_f^s = \min(r \in R : F_r = f) P_r^s, \forall f \in F \quad (22)$$

$$O_f^e = \max(r \in R : F_r = f) P_r^e, \forall f \in F \quad (23)$$

$$Prec_{r_2} = r_1 \Rightarrow D_{r_1}^e + D[M_{r_1}, F_{r_2}] \leq P_{r_2}^s, \forall r_1 \in I, \forall r_2 \in O \quad (24)$$

$$Prec_{r_2} = r_1 \Rightarrow W_{r_2}^f = P_{r_2}^s - D_{r_1}^e - D[M_{r_1}, F_{r_2}], \forall r_1 \in R \forall r_2 \in R \quad (25)$$

$$P_r^e + D[M_r, F_r] \leq D_r^s, \forall r \in R \quad (26)$$

$$W_r^m = D_r^s - P_r^e - D[M_r, F_r], \forall r \in R \quad (27)$$

$$\text{ALLDIFFERENT}(\text{Prec}) \quad (28)$$

$$Prec_r \neq r, \forall r \in R \quad (29)$$

$$D_r \prec H, \forall r \in R \quad (30)$$

$$H^e \leq T, \quad (31)$$

All constraints of the CP model have been expressed in the same way as in the CBLs model except constraints (20) and (21) that express that loading and unloading require a log-loader for their execution, and constraint (30) that represents the fact that all requests are served before the dummy task (H).

5.2.3 CP Search Strategy

We used COMET2.0 CP solver library to define the search strategy, which consists in scheduling the unloading activities as soon as possible. This is accomplished by using the procedure *setTimes* that assigns the starting time of the activities given in argument. The search continues

by generating appropriate values for the truck waiting time variables using the procedure *label* which performs a non-deterministic assignment of values to integer variables given in argument. The search space is explored using a depth-first strategy (DFS), which allows to find feasible solutions quickly compared to other strategies. This is important, since the hybridization imposes the best solution found so far as an upper bound in the CP model. Thus, any feasible solution of the CP part outperforms the best CBLS solution. We limited the global computational time allocated to CP and imposed a restarting strategy after a fixed amount of time. This is accomplished by using the procedure *restartOnTimeLimit*.

5.2.4 The Hybrid Algorithm

We propose a hybrid ILS/CP algorithm to solve the daily SLTSP. The idea behind using CP is to measure the efficiency of the greedy algorithm compared to the COMET2.0 CP solver, and to improve the solution quality of the greedy algorithm when possible. Since CP consumes more time than the greedy algorithm, we have investigated its use in five possible situations, which have been tested independently.

H_1 : Call the CP model when all the ILS moves are unable to improve the best solution found so far.

H_2 : Call the CP model when the current ILS move improves the routing and scheduling part of the best solution found so far.

H_3 : Use the CP model when the current ILS move improves only the routing part of the best solution.

H_4 : Use the CP model only when the scheduling part of the best solution has been improved by the current ILS move.

H_5 : Perform the hybridization when the current ILS move deteriorates the routing and scheduling part of the best solution found so far.

An ILS move represents a combination of a neighborhood move followed by a greedy scheduling. We note that for all these cases the best solution found so far has been imposed as an upper bound of the objective function in the CP model.

Figure 1: The second phase hybrid algorithm

In practice, combining the initial assignment ($Prec_r = r - nbV, \forall r \in O$) and the neighborhood structures $\{N_1, N_2, N_3\}$ allows to remove constraints (16), (17), (28) and (29) from the CBLIS and CP models respectively. For more details about the ILS and ILS/CP, see Figure 1.

6 Experimental Results

We have tested our algorithmic approach on two different industrial case studies provided by the FERIC division¹ of FPIInnovations, both of them involving six forest areas and five woodmills. For the first case study, the average cycle time to transport a load is about 4 hours and there are approximately 400 loads per week, while the second case has an average cycle time of 5.5 hours and approximately 700 loads per week. In both cases, there are three different products to transport from forest areas to woodmills, and we used the following costs: \$60 per hour for the waiting time of trucks, \$70 per hour for truck travel, and \$100 per hour for the waiting time of log-loaders. For both cases, loading and unloading takes about 20 minutes, which motivated our choice to discretize the data in 20 minute-increments, although the loading takes slightly

¹The Forest Engineering Research Institute of Canada is a private, not-for-profit research and development organization whose goal is to improve Canadian forestry operations related to the harvesting and transportation of wood, and the growing of trees, within a framework of sustainable development.

more time.

We first report that the tactical phase can be solved to optimality for the two instances in less than fifteen minutes using only the default search procedure of SCIP. The total cost related to this phase and the observed computing time are given in Table 1.

For the daily phase, we also conducted tests to assess the contribution of each neighborhood to the quality of the final solution. We counted the number of times that a neighborhood improved the current best solution. These results are reported in Table 2 for the pure ILS algorithm; they show that the three neighborhoods contribute effectively to improve the best solution, even if the last neighborhood seems to be somewhat less effective than the others. However, a lower improvement rate can be expected from N_3 , as this neighborhood is only called when the previous ones are unable to escape a local minimum. During these tests, no significant difference in the contribution of neighborhoods were observed between the pure and the hybrid versions.

Cases	Total cost	Time
1	23,590\$	793 seconds
2	61,110\$	5 seconds

Table 1: **The tactical phase results**

Cases	N_1	N_2	N_3
1	53%	36%	13%
2	49%	33%	18%

Table 2: **Effectiveness of the neighborhoods**

Several tests were performed by varying the number of trucks, the time allocated to the CP procedure and, finally, the global time allocated to solve the daily Synchronized LTSP. We ran each scenario thirty times and according to our preliminary tests, the first approach H_1 , which integrates CBLS and CP when all the ILS moves are unable to improve the best solution found so far, outperforms all other hybrid approaches. We thus choose to compare the pure ILS algorithm (without hybridization) with this first hybrid approach. We have run the hybrid algorithm allowing the CP procedure to run for 20, 40 or 60 seconds. A comparison of the results obtained indicated that the best value to use was 40 seconds for all scenarios. In the following, we only consider the results obtained for this value.

To assess the novelty of the method presented in this paper, we decided to compare this approach with a previous one (IP/CP, see [8]) that only addressed the daily problems. For both cases, we selected three daily problems arising from the tactical phase of our new approach, and compared the best solution of [8] to the best and the average solution of the new approach (see Table 3). These results clearly show that the proposed method (ILS/CP) outperforms the previous one (IP/CP), since for both cases studies, both the average and the best solutions of ILS/CP are better than the best solution of IP/CP. We note that for case study 1, the difference between the results obtained by the two methods was considerable (about 20%), whereas for the other case, it varies between 1% and 9%.

We report in Tables 4 and 5 the average total cost, the average routing cost, the average waiting time cost and the worst waiting time cost obtained in 30 runs by the pure ILS method and the ILS/CP for fixed computing times. Even if CP is time consuming, since it takes 40 seconds at each non-profitable global iteration of the hybrid algorithm, we note that for both case studies, the two methods are almost equivalent in terms of average total cost. For the eighteen scenarios, the ILS/CP method performs better than the pure ILS for ten scenarios, whereas the opposite is true for the remaining scenarios. This is explained by the fact that the waiting time cost represents less than 7% of the total cost. Therefore, the CP contribution has an impact on only a small fraction of the overall objective function. However, in terms of average waiting time cost, ILS/CP outperforms ILS in more than two thirds of scenarios by margin gains from 1% to 14%. The same observation can be made for the worst waiting time cost, since again, we note that ILS/CP performs better than ILS for almost all scenarios, while in terms of average routing cost there is no significant difference between the results of the two approaches. This reduction in terms of the average waiting cost and worst waiting cost is due to the contribution of CP, which is specially well-adapted to deal with the synchronization constraints.

The introduction of CP (ILS/CP) allows us to obtain more stable results and therefore increases the robustness of the algorithm, in the sense of having less variance in the total cost (see Table 7). The convergence of both approaches is illustrated in Figures 2 and 3. We notice that pure ILS converges more rapidly to the best solution compared to the hybrid approach ILS/CP. This is explained by the fact that using CP in the ILS/CP approach requires solving the CP model, which takes 40 seconds for each call, while the other approach starts immediately the search by perturbing the current best solution. Pure ILS provides better solutions than ILS/CP for small computational times (less than 2 minutes in our cases). However, our objective is to optimize daily plans for a complete week and small computational times are not mandatory in such a situation. As we can see from Table 6, ILS/CP is able to identify the best solution

obtained during the 30 runs in most cases when the overall computational is limited to 10 or 15 minutes. However, the opposite is true when the computational time is set to 20 minutes.

We believe that all these advantages (improvement of average waiting time cost, improvement of the variance and reduction of the worst scenarios) are the result of the combination of ILS and CP. Even if, both algorithms have a similar global behavior, the fact that ILS/CP obtains smaller waiting times encourages us to use it, since this part of the solution is directly perceived by the drivers.

Cases	Best IP/CP	ILS/CP	
		Best Sol	Average Sol
Case 1	3423	2563	2680
	4160	3516	3613
	4230	3413	3446
Case 2	12353	12003	12236
	12593	12020	12240
	12323	11023	11273

Table 3: **Approach comparison with one hour of computation time**

Cost	nbV	Global time 10 min		Global time 15 min		Global time 20 min	
		ILS	ILS/CP	ILS	ILS/CP	ILS	ILS/CP
Average total	14	54596	54490	54143	54190	53950	53960
	15	54276	54090	53990	53996	53930	53870
	16	53690	53486	53516	53350	53390	53343
Average routing	14	51000	50926	51033	51193	51313	50836
	15	50993	50693	50813	51006	50670	50660
	16	50796	50700	50616	50513	50586	50580
Average waiting	14	3593	3563	3080	2996	2636	3120
	15	3293	3396	3173	2990	3256	3206
	16	2893	2786	2896	2840	2803	2760
Worst waiting	14	4446	4153	4060	4060	3266	3210
	15	3526	3500	3966	3800	4066	3900
	16	3126	3606	3213	3140	3106	3246

Table 4: **Results of case 1**

7 Conclusion

We have presented the weekly log-truck scheduling problem in which one seeks to minimize the sum of transportation and unproductive waiting time costs. To address this problem, we

Cost	nbV	Global time 10 min		Global time 15 min		Global time 20 min	
		ILS	ILS/CP	ILS	ILS/CP	ILS	ILS/CP
Average total	30	139496	139713	139350	139103	138856	138686
	31	139536	140203	139403	138533	138813	138526
	32	139390	141113	139000	139973	138883	139286
Average routing	30	133536	133903	133400	133280	133426	133293
	31	133590	134413	133683	133076	132100	133530
	32	133706	135076	133470	134206	133600	133750
Average waiting	30	5960	5809	5946	5823	5426	5393
	31	5943	5790	5720	5453	5710	4993
	32	5680	6036	5526	5763	5283	5533
Worst waiting	30	6333	6040	7033	6046	5740	5980
	31	6400	6280	6200	5866	5953	5400
	32	6433	6306	6093	6020	5760	6046

Table 5: **Results of case 2**

Case	nbV	Global time 10 min		Global time 15 min		Global time 20 min	
		ILS	ILS/CP	ILS	ILS/CP	ILS	ILS/CP
Case1	14	54203	54170	53300	53520	53446	53630
	15	53920	53430	53480	53430	53460	53400
	16	53120	53043	53156	53093	52550	52896
Case2	30	138700	137883	137686	137520	136786	137686
	31	138980	139550	137753	137636	136513	137730
	32	138486	139966	138223	139163	137743	138536

Table 6: **The best total cost results**

Time(min)	Total cost variance (%)			
	case 1		case 2	
	ILS	ILS/CP	ILS	ILS/CP
10	0.4	0.4	0.6	0.6
15	0.8	0.7	0.7	0.5
20	0.7	0.6	0.4	0.3

Table 7: **The variance results**

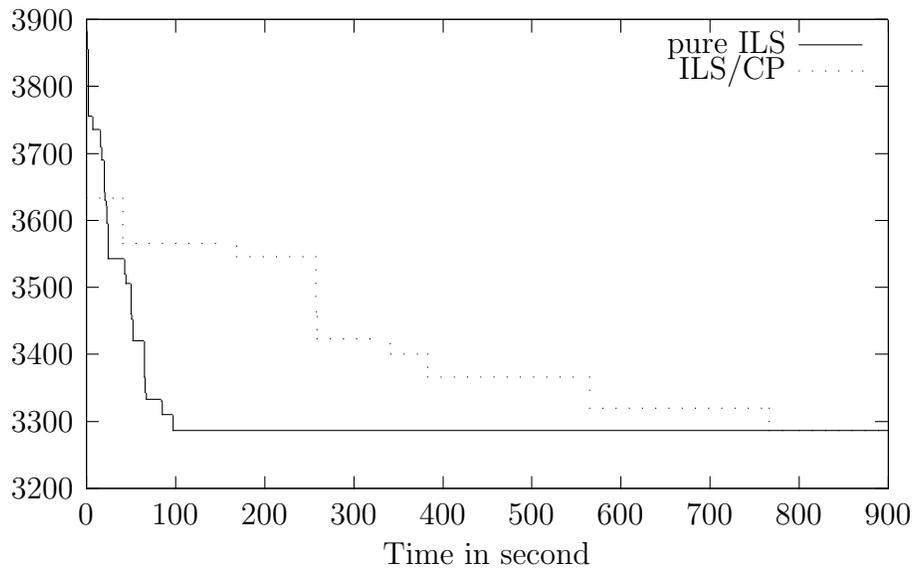


Figure 2: the convergence of the algorithm : case 1

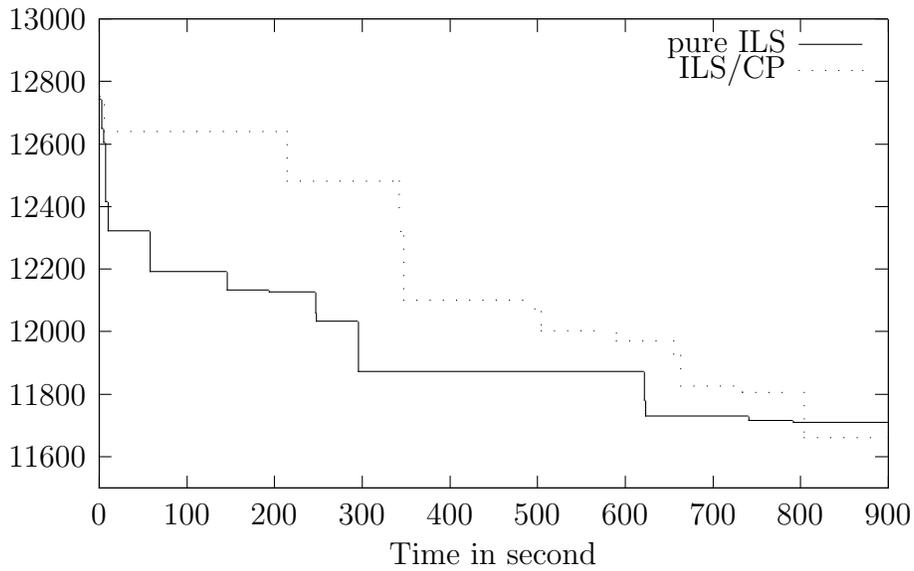


Figure 3: the convergence of the algorithm : case 2

decomposed it into two phases. The first phase determines the destination of full truck loads (from forest areas to woodmills) by minimizing the transportation cost. This phase yields seven daily problems covering the whole week, which are then solved sequentially. The objective of this second phase is to schedule the daily transportation of logs by minimizing the cost of unproductive waiting time trucks and log-loaders. As the second phase includes a strong scheduling component, we have investigated the use of both a constraint-based local search and a constraint programming approaches. The results indicate that using a CP model within the heuristic allows for slightly better results, especially in terms of waiting times, while improving the overall robustness of the approach.

Future research directions involve solving phase 2 using integer programming software and dealing with additional constraints such as the home bases of trucks, the scheduling of daily lunch breaks and supply constraints.

References

- [1] Andersson, G., P. Flisberg, B. Lidén, and M. Rönnqvist. RuttOpt – a decision support system for routing of logging trucks. *Canadian Journal of Forest Research*, 38:1784–1796, 2008.
- [2] Baptiste, P., C. LePape, and W. Nuijten. Constraint-Based Optimization and Approximation for Job-shop Scheduling. In *Proceedings of AAAI-SIGMAN Workshop on Intelligent Manufacturing Systems*, pp. 5–16, IJCAI’95, Montreal, Canada, 1995.
- [3] Baptiste, P., C. LePape, and W. Nuijten. *Constraint-Based Scheduling*. Kluwer Academic Publishers, 2001.
- [4] Bredström, D., and M. Rönnqvist. Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. *European Journal of Operational Research*, 191(1):19–31, 2008.
- [5] Bräysy, O. A reactive variable neighborhood search for the vehicle routing problem with time windows. *INFORMS Journal on Computing*, 15:347–368, 2003.
- [6] Cordeau, J-F., G. Desaulniers, J. Desrosiers, M.M. Solomon, and F. Soumis. VRP with time windows. In: P. Toth and D. Vigo (eds.), *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, vol 9, Philadelphia, PA, pp. 157–193, 2002.
- [7] Dumas, Y., J. Desrosiers, and F. Soumis. The pickup and delivery problem with time windows. *European Journal of Operational Research*, 54:7–22, 1991.

- [8] El Hachemi, N., M. Gendreau, and L-M. Rousseau. A Hybrid Constraint Programming Approach to the Log-Truck Scheduling Problem. *Annals of Operational Research*, 10.1007/s10479-010-0698-x.
- [9] Everborn, P., P. Flisberg, and M. Rönnqvist. Laps care- and operational system for staff planning of home care. *European Journal of Operational Research*, 171(3): 962–976, 2006.
- [10] Everborn, P., M. Rönnqvist, H. Einarsdóttir, M. Eklund, K. Lidén, and M. Almroth. Operations Research Improves Quality and Efficiency in Home Care. *Interfaces*, 39(1): 18–34, 2009.
- [11] Freling, R., D. Huisman, and A.P.M. Wagelmans. Models and algorithms for integration of vehicle and crew scheduling. *Journal of Scheduling*, 6:63–85, 2003.
- [12] Flisberg, P., B. Lidén, and M. Rönnqvist. A hybrid method based on linear programming and and tabu search for routing of logging trucks. *Computers and Operations Research*, 36(4):1122–1144, 2008.
- [13] Gendreau, M., G. Laporte, and J-Y. Potvin. Metaheuristics for the Capacitated VRP. In: P. Toth and D. Vigo (eds.), *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, vol 9, Philadelphia, PA, pp. 129–154, 2002.
- [14] Gronalt, M., and P. Hirsch. Log-Truck Scheduling with Tabu Search Strategy. *Metaheuristics*, 39:65–88, 2007.
- [15] Haase. K., G. Desaulniers, and J. Desrosiers. Simultaneous vehicle and crew scheduling in urban mass transit systems. *Transportation Science*, 35(3):286–303, 2001.
- [16] Hane C., C. Barnhart, E.L. Johnson, R. Marsten, G. Nemhauser, and G. Sigismondi. The fleet assignment problem: Solving a large-scale integer program. *Mathematical Programming*, 70:211–232, 1997.
- [17] Van Hentenryck, P. *Constraint Satisfaction in Logic Programming*. MIT Press, 1989.
- [18] Ibaraki, T., S. Imahori, K. Nonobe, K. Sobue, T. Uno, and M. Yagiura. An iterated local search algorithm for the vehicle routing problem with convex time penalty functions. *Discrete Applied Mathematics* (156):2050–2069, 2008.
- [19] Ioachim, J., J. Desrosiers, F. Soumis, and N. Bélanger. Fleet assignment and routing with schedule synchronization constraints. *European Journal of Operational Research* (119):75–90, 1999.

- [20] Karlsson, J., M. Rönnqvist, and J. Bergström. An optimization model for annual harvest planning. *Canadian Journal of Forest Research*, 34(8):1747–1754, 2004.
- [21] Linnainmaa, S., J. Savalo, and O. Jokinen. EPO: A knowledge based system for wood procurement management. Paper presented at the *7th Annual Conference on Artificial Intelligence*, Montreal, 1995.
- [22] Michel, L., and P. Van Hentenryck. A modeling language for local search. In *Proceedings of the Third International Conference on the Principles and Practice of Constraint Programming (CP'97)*, pp. 237–251, 1997.
- [23] Michel, L., and P. Van Hentenryck. Localizer. *Constraints*, 5:41–82, 2000.
- [24] Murphy, G. Reducing trucks on the road through optimal route scheduling and shared log transport services. *Southern Journal of Applied Forestry*, 27(3):198–205, 2003.
- [25] Palmgren, M., M. Rönnqvist, and P. Värbrand. A solution approach for log truck scheduling based on composite pricing and branch and bound. *International Transactions in Operational Research*, 10:433–447, 2003.
- [26] Palmgren, M., M. Rönnqvist, and P. Värbrand. A near-exact method for solving the log-truck scheduling problem. *International Transactions in Operational Research*, 11:447–464, 2004.
- [27] Potvin, J.-Y., T. Kervahut, B.L. Garcia, and J-M. Rousseau. The vehicle routing problem with time windows, Part 1: tabu search. *INFORMS Journal on Computing*, 8:158–164, 1996.
- [28] Rönnqvist, M. Optimization in forestry. *Mathematical Programming*, 97:267–284, 2003.
- [29] Rönnqvist, M., H. Sahlin, and D. Carlsson. *Operative planning and dispatching of forestry transportation*. Report LiTH-MAT-R-1998-18. Linköping University, Linköping, Sweden, 1998.
- [30] Rönnqvist, M., and D. Ryan. Solving truck dispatch problem in real time. In *Proceedings of the 31st Annual Conference of the Operational Research Society of New Zealand*, 31 August - 1 September 1995, Wellington, New Zealand. The Operational Research Society of New Zealand, Auckland, N.-Z., pp. 165–172.
- [31] Ropke, S., J.-F. Cordeau, and G. Laporte. Models and Branch-and-Cut Algorithm for Pick-up and Delivery Problems with Time Windows. *Networks*, (49),4:258–272, 2007.

- [32] Rousseau, L.-M., M. Gendreau, and G. Pesant. The Synchronized Vehicle Dispatching Problem. In *Proceedings of Odysseus*, Palermo (Italy), 3 pages, 2003.
- [33] Savelsbergh, M.W.P., and M. Sol. The general pickup and delivery problem. Available on the web.
- [34] Taillard, E., P. Badeau, M. Gendreau, F. Guertin, and J-Y. Potvin. Tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, 31: 170–186, 1997.
- [35] Toth, P., and D. Vigo. The Vehicle Routing Problem. Society for Industrial and Applied Mathematics, 2001.
- [36] Van Hentenryck, P., and L. Michel. Constraint-Based Local Search. MIT Press, 2005.
- [37] Van Hentenryck, P., and L. Michel. Differentiable Invariants. In *Proceedings of the Twelfth International Conference on Principles and Practice of Constraint Programming (CP'06)*, pp. 685–689, 2006.
- [38] Weintraub, A., R. Epstein, R. Morales, J. Seron, and P. Traverso. A truck scheduling system improves efficiency in the forest industries. *Interfaces* (26),4:1–12, 1996.
- [39] Zweben, M., and M. Fox. *Intelligent Scheduling*. Morgan Kaufman, 1994.