# A Hybrid Constraint Programming Approach to the 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.** Scheduling problems in the forest industry have received significant attention in the recent years and have contributed many challenging applications for optimization technologies. This paper proposes a solution method based on constraint programming and mathematical programming for a log-truck scheduling problem. The problem consists of scheduling the transportation of logs between forest areas and woodmills, as well as routing the fleet of vehicles to satisfy these transportation requests. The objective is to minimize the total cost of the non-productive activities such as the waiting time of trucks and forest log-loaders and the empty driven distance of vehicles. We propose a constraint programming model to address the combined scheduling and routing problem and an integer programming model to deal with the optimization of deadheads. Both of these models are combined through the exchange of global constraints. Finally the whole approach is validated on real industrial data.

## 1 Introduction

The forest industry occupies an important place in the economy of several countries such as Chile, Canada, Sweden, Finland and New Zealand. Planning problems in forestry cover a wide scope of activities ranging from planting and harvesting to road building and transportation. Furthermore, in most problems, it is critical to pay attention to important environmental issues, as well as to company-specific goals and operating rules.

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. Since the average distance between forest areas where wood is collected and mills to which this wood is transported is around 150 km, more than 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. It thus follows that transport activities between forest areas and mills should be organized as effectively as possible, both for economic and environmental reasons.

### 1.1 The Log-Truck Scheduling Problem

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" (see, for instance, Ropke *et al.* [20]). In our case, we consider a pick-up and delivery problem in which for each request exactly one truckload 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, since all requests are also assumed to be known in advance.

It must be noted that the problem as described above differs in several ways from the LTSPs addressed by some other authors. These differences stem from specific characteristics of Canadian forestry operations. Among others, 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 same destination. As a result, there is no need to explicitly take into account the several products (about 3 products in our case), since they are implictly included in the full truckloads. While quantities are generally expressed in cubic meters, these can be easily converted into truckloads.

In this paper, we assume that we have predetermined destinations (a set of fixed transportation requests) as in Gronalt and Hirsh [8]. In our case, the daily transportation requests are derived from decisions made at the weekly planning level.

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 the mills and we are allowed to reallocated trucks among the mills to obtain more efficient solutions. Driver changes are normally performed at mills 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 been taken into account; these will be the subject of future work.

Furthermore, in 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, then 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.

### 1.2 Litterature Review

Since the mid-1990s, several companies in the forestry sector have initiated major projects aimed at improving the transportation portion of their activities, in particular, the control and quality of truck scheduling.

Weintraub *et al.* [28] describe a heuristic-based model (ASICAM) that produces a daily plan for each truck by assigning loads and trips, while satisfying supply and demand constraints. 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.

Another system called EPO developed by Linnainmaa *et al.* [11] has been used in Finland. EPO is a system that deals with all stages from strategic to operative planning. The input data is collected on-line directly from the forest areas and the main output is a weekly schedule for each truck. A main goal of EPO is the minimization of truck driving as whole.

Palmgren *et al.* [15] describe a near-exact method for solving the LTSP. This approach is based on column generation and pseudo-branch-and-price, where each column represents one feasible route for one truck. An initial set of routes is generated at the beginning. Then, the subproblem, which is a constrained shortest path problem, is solved by applying a k-shortest path algorithm. The columns whose reduced cost is negative will be communicated to the master problem.

Gronalt and Hirsch [8] apply a modified version of the Unified Tabu Search heuristic to solve the LTSP where they vary the size of the neighborhoods in an oscillating fashion.

Flisberg *et al.* [7] and Andersson *et al.* [1] propose a two-phase solution approach that transforms the LTSP into a standard vehicle routing problem with time windows. The first phase determines the flow of wood from supply points to demand points, while the second phase combines transport nodes into routes. The dispatching procedure that continuously updates the trucks routes during the day is based on the previous work of Rönnqvist and Ryan [18] and Rönnqvist *et al.* [19].

With respect to numerical experiments, Murphy [13] presents a case study with on average 9 trucks and 35 transport tasks per day, while Palmgren *et al.* [14] have solve 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 [8] solve random problems with 30 transport tasks and 10 trucks. Finally, Flisberg *et al.* [7] and Andersson et al [1] have solve substantially larger instances ranging from 188 transport tasks to about $2,500$ full truckloads with 15 to 110 trucks.

For a more detailed description of optimization problems in the forest sector, we refer the reader to Rönnqvist [17].

There are many applications in which vehicles must be synchronized. This occurs in distribution systems when a vehicle cannot pick up a load until another vehicle has first delivered it, or in urban mass transit systems when drivers have to changebuses at so-called relief points (see Freling et al [6] and Haase et al [9]). Bredström and Rönnqvist [2] present a mathematical programming model for the combined vehicle routing and scheduling problem with additional precedence and synchronization constraints. To illustrate the practical significance of the temporal precedence and synchronization constraints, the authors test their approach with the homecare staff scheduling problem and discuss the importance of synchronizaton constraints in the airline industry. They also discuss the need of coordination between harvesters and forwarders in the forestry context. Since forwarding can only be done once the harvesting has been performed. With respect to the LTSP, the authors discuss the requirement to synchronize trucks (without crane) and loaders, since these loaders often serve several stands and move between these in order to load trucks.

### 1.3 A Constraint Programming Approach

Constraint Programming (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 Baptiste *et al.* [3, 4]; Van Hentenryck [25]; Zweben and Fox [29]). In the context of the LTSP, it is a particularly attractive approach, because it can easily model both the synchronization of trucks and log loaders, and the optimization of the waiting costs.

Combining constraint programming and linear or mixed integer programming (MIP) has received a lot of attention since the late 1990's. One of the simplest approach consists in defining a subproblem that can be optimized through linear programming (LP) or MIP solver and whose optimal solution can be used by a CP master model (Sakkout and Wallace [22]). Other forms of hybridization have been proposed for the Dantzig-Wolfe decomposition (Fahle and Sellmann [5]; Sellmann *et al.* [23]; Rousseau *et al.* [21] ); these take advantage of the flexibility of constraint programming to generate columns (decision variables) in a branch-and-price framework in the context of vehicle routing and crew rostering applications. Benders decomposition has been investigated by Hooker [10] who combines MIP and CP to solve planning and scheduling problems: the MIP allocates tasks to facilities, while the scheduling is performed by CP; the two are linked via logic-based Benders decomposition.

In a context similar to forest transportation (food industry transportation), Simonis *et al.* [24] developed the TACT program whose solution approach consist of three steps: the first one determines the number of trips between farms and factories and the different truck types by solving an IP model; the second step takes these trips and schedules them in time; finally, the third step assigns individual resources to all activities. For more details about decomposition methods involving CP, we refer the reader to Milano [12].

This paper presents a CP model for the LTSP based on an integer programming (IP) subproblem that first minimizes the total distance of deadhead trips. The solution of this IP is then used to generate global cardinality constraints for the CP scheduling model. To our knowledge this is the first time that the problem of synchronizing of trucks and log loaders is explicitly modeled and solved. Moreover, the contribution of this paper also lies in the communication between IP and CP through the use of structured global constraints, which we believe is novel.

The paper is organized as follows. Sections 2 and 3 present respectively the basic CP model and the solution approaches that we developed for solving the LTSP. The experimental setting is described in Section 4, where computational results are also reported. Section 5 concludes the paper.

## 2 Problem Description and Model

We now describe a CP model for the LTSP. CP presents many advantages in this context, such as allowing to easily express the difficult constraints (alternative assignment, routing) and to synchronize the trucks and the log loaders. The model is based on the scheduling language introduced by Van Hentenrick [26] where problems are stated in terms of activities and ressources. Note that a unary resource is a resource that cannot be shared by two activities (i.e., as soon as an activity requires the resource, no other activities can use of that resource). Alternative resources are sets of unary resources that are equivalent from the activity standpoint.

### 2.1 Parameters

| | |
|---|---|
| $D[m, f]$ | : distance between mill $m$ and forest area $f$ $(= D[f, m])$, |
| $nbR$ | : number of (real) transportation requests, |
| $nbV$ | : number of vehicles, |
| $R$ | : set of real transportation requests, |
| $I$ | : $R$ augmented with dummy nodes representing the original locations of trucks, |
| $O$ | : $R$ augmented with dummy nodes reprensenting the final destinations of trucks, |
| $F$ | : set of forest areas, |
| $F_r$ | : forest origin of request $r$, |
| $M$ | : set of woodmills, |
| $M_r$ | : woodmill destination of request $r$, |
| $V$ | : set of vehicles defined as *alternative unary resources*, |
| $L^m$ | : log loader at mill $m$ defined as a *unary resource*, |
| $L^f$ | : log loader at forest area $f$ defined as a *unary resource*, |
| $T$ | : deadline for transporting all requests, |
| $W$ | : domain of waiting time variables (0..T), |
| $c_t$ | : cost of waiting one hour for a truck, |
| $c_l$ | : cost of waiting one hour for a log-loader, |
| $c_d$ | : hourly cost of driving an empty truck. |

### 2.2 Variables and domains

The decision variables of the problem are based on the ILOG Scheduler component of OPL Studio 3.7 (see Van Hentenryck [26]). 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.

| | |
|---|---|
| $H$ | : *activity* of duration 0 that proceeds after all other activities, it constitutes the time horizon of the plan; |
| $C_r, \forall r \in R$ | : combined *activity* of loading, traveling and unloading request $r$; |
| $P_r, \forall r \in R$ | : pick-up *activity* of duration $d^p$ associated with request $r$; |
| $D_r, \forall r \in R$ | : delivery *activity* of duration $d^d$ associated with request $r$; |
| $V_r \in V, \forall r \in R$ | : vehicle assigned to request $r$; |
| $S_r \in O, \forall r \in I$ | : successor of request $r$ on the same vehicle; |
| $O_f, \forall f \in F$ | : activity representing the opening time of forest area $f$; |
| $W_r^f \in W, \forall r \in R$ | : waiting time of a truck at the forest area of request $r$; |
| $W_r^m \in W, \forall r \in R$ | : waiting time of a truck at the woodmill of request $r$; |
| $W_l^f \in W, \forall f \in F$ | : waiting time of the log-loader in the forest area $f$. |

### 2.3 Constraints and Objective Function

Since we consider fixed transportation requests, several quantities are independant of the solution and can thus be removed from the objective function: these are the cost related to the loaded transportation distance as well as to the loading and unloading activities. Therefore it is sufficient to minimize the cost of non-productive activities, i.e. the waiting time of trucks and forest log-loaders and the empty driven distance of vehicles. Several reasons motivate this choice such as reducing the greenhouse gas emission and increasing the trucks productivity. In some variants of the LTSP studied in the litterature, loaders work a fixed shift, 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 maximise their usage.

$$\text{Minimize} \sum_{r \in R} c_t(W_r^m + W_r^f) + \sum_{f \in F} c_l W_l^f + c_d \sum_{r \in R} D[M_r, F_{S_r}]$$

subject to

$$C_r \rightleftharpoons V,\ \forall r \in R \quad (1)$$

$$P_r \rightleftharpoons L^{F_r},\ \forall r \in R \quad (2)$$

$$D_r \rightleftharpoons L^{M_r},\ \forall r \in R \quad (3)$$

$$C_r^s = P_r^s,\ \forall r \in R \quad (4)$$

$$C_r^e = D_r^e,\ \forall r \in R \quad (5)$$

$$S_r \neq r,\ \forall r \in R \quad (6)$$

$$S_{r_1} = r_2 \Rightarrow D_{r_1}^e + D[M_{r_1}, F_{r_2}] \leq P_{r_2}^s,\ \forall r_1 \in I,\ r_2 \in O \quad (7)$$

$$V_{S_r} = V_r,\ \forall r \in I \quad (8)$$

$$\textsc{alldifferent}(S) \quad (9)$$

$$\text{AHSR}(C_r, V, v) \Leftrightarrow V_r = v,\ \forall r \in I,\ \forall v \in V \quad (10)$$

$$P_r^e + D[F_r, M_r] = W_r^m + D_r^s,\ \forall r \in R \quad (11)$$

$$S_{r_1} = r_2 \Rightarrow W_{r_2}^f = P_{r_2}^s - D_{r_1}^e - D[M_{r_1}, F_{r_2}],\ \forall r_1,\ r_2 \in R \quad (12)$$

$$O_f^e = max(P_r^e | r \in R : F_r = f),\ \forall f \in F \quad (13)$$

$$O_f^s = min(P_r^s | r \in R : F_r = f),\ \forall f \in F \quad (14)$$

$$W_l^f = O_f^e - O_f^s - \sum_{r \in R : F_r = f} d^p,\ \forall f \in F \quad (15)$$

$$C_r \prec H,\ \forall r \in R \quad (16)$$

$$H^e \leq T \quad (17)$$

Constraint (1) expresses the fact that each request *requires* ($\rightleftharpoons$) a truck for its execution. Constraints (2) and (3) express the fact that loading and unloading *require* a log loader for their execution. Constraints (4) and (5) mean that the beginning time of a request must coincide with the beginning of its associated loading and that the ending time of a request coincides with the end of its associated unloading. Constraint (6) specifies that the successor of a request cannot be the request itself. Constraint (7) ensures that a truck that carries out two successive requests (including the dummy departures and ending requests) has enough time to do so. Constraint (8) expresses the fact that a request and its successor are served by the same truck. Constraint (9) ensures that all successors are given a different value (this, combined with the fact that a successor variable can only have one value, is equivalent to a flow conservation constraint). Constraint (10) is an OPL constraint (*ActivityHasSelectedResource*) that connects the transportation and scheduling component of the CP model (constraint $\mathit{ActivityHasSelectedResource}(a, S, u)$ holds if activity $a$ has selected resource $u$ in the set of alternative resources $S$). Constraints (12), (11) and (15) compute the waiting time of trucks at forest areas, woodmills and the forest log-loaders waiting time. Constraints (13) and (14) ensure that the activity $O_f$ has a duration equal to the opening time of forest area $f$. Constraint (16) specifies that each request must be executed before the makespan dummy task ($H$) (it precedes it ($\prec$)). The constraint (17) ensures that we respect the deadline to transport all the requests.

## 3 Solving the LTSP

We solved the proposed model using the Ilog *OPL-Studio* suite based on *Scheduler*, a specialized constraint programming library for modeling scheduling problem.

The search strategy consists of two steps. In the first step, we route the vehicles with the procedure *generate* of OPL Studio 3.7, which identifies a value for each successor variable $S_r, \forall r \in I,$. This instruction recursively identify the variable, which has the smallest domain and tries to assign it values in a lexicographic order. Since the values are ordered so that requests on any given forest site have consecutive numbers, this search strategy sends all trucks to first site until all the requests have been serviced, which tends to minimize the opening time of a forest site and thus the unproductive time of its loader. We have implemented several other approaches, which aimed at reducing the deadheads or the waiting time of the trucks, but the solutions they produced showed a unacceptably low usage of forest loaders. In the second step, once the successor variables are fixed, we schedule each request as soon as possible by employing the built-in scheduling algorithms of Ilog Scheduler (*setTimes* and *rankGlobal*).

Finally, we explore the solution space using a dichotomic search procedure on the total cost and a depth-first search strategy (DFS) on all branching decisions (both for the routing and scheduling part). For a more detailed description of strategies, we refer the reader to Van Hentenryck *et al.* [27].

### 3.1 Decomposing the LTSP

Since the cost of transportation is mainly based on distance, it is crucial to minimize the unproductive distance driven by empty vehicles. However, since the chosen search strategy does not attempt to reduce the traveled distance, we propose a decomposition approach where this criteria is explicitly adressed. We model the circulation of trucks between the mills and the forest areas as a network flow problem with some additional constraints that can be easily solved as an integer program (IP). This model yields an optimal solution with respect to the deadhead component of the objective function, but it is, however, not able to schedule the trucks and the log-loaders. To link the IP and CP models, we have considered three different approaches: 1) Solve the IP, keep the optimal objective value, and use it as a constraint on the deadhead component of the objective function in the CP model; 2) Solve the IP, keep the optimal solution, and use it to fix successor variables in the CP model; 3) Solve the IP, look at the structure of the optimal solution, and impose it in the CP model through the introduction of global constraints on successor variables.

Since the objective function of the CP model is basically a large sum of small elements, it unfortunately does not allows for good back propagation (bounding the objective does not really trigger propagation and domain reduction). For that reason, method 1 would not be very useful in our context. On the other hand, since the IP model completely ignores the important scheduling aspects of the problem, fixing the sequence of all requests in the CP model would overly constrain the scheduler. Once the complete sequence is given, there is not enough flexibility left to avoid important waiting times. The challenge is thus to identify a good solution to the deadhead problem, while still giving the CP model enough flexibility to minimize waiting times in the final schedule.

For this reason we chose to migrate from the IP model only the minimal information that would allow achieving the minimal deadhead value. Since all full loads must be transported from their destination, we observed that the optimal value is completely determined by the arcs representing empty trips in the solution. It is thus not the global sequence that is important, but rather the number of empty trips performed between each mill and forest site.

These numbers can be extracted from the IP optimal solution and imposed in the CP model through the introduction of Global Cardinality Constraints (GCC, see Régin [16]). We thus constrain the CP model to use the correct number of deadhead trips between each mill-forest pair. Imposing this structure considerably reduces the search space and speeds up the resolution.

### 3.2 IP Model, variables, parameters and constraints

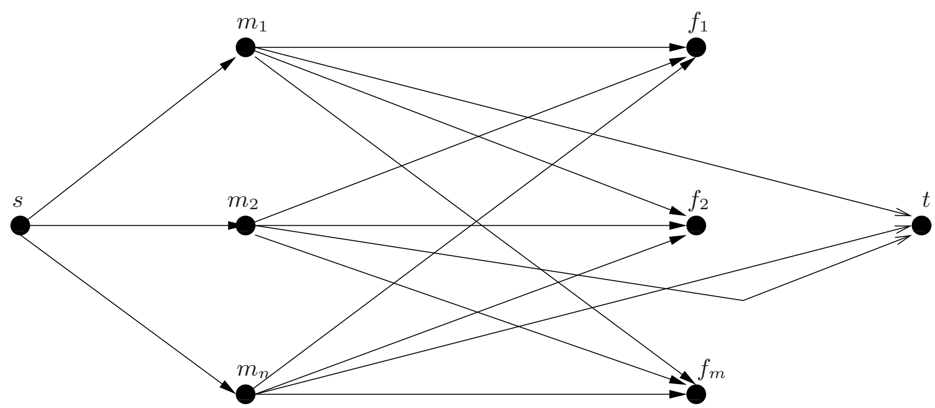


**Fig. 1.** Graph associated with the empty driving

In the Figure 1, $s$ and $t$ are two dummy nodes. The nodes on the left- and right-hand side represent respectively woodmills and forest areas. For each mill, we associate a demand that must be fulfilled, and each forest area provides a supply sufficient to meet the requests.

$x_{ij}$ : number of empty trucks driving from woodmill $i$ to forest area $j$,
$s_i$ : number of trucks that start at woodmill $i$,
$e_i$ : number of trucks that end up at woodmill $i$,
$d_i$ : number of requests associated with woodmill $i$,
$p_j$ : number of loads associated with forest area $j$.

The objective function of this IP model consists in minimizing the deadhead.

$$\text{Minimize } \textstyle\sum_{i \in M} \sum_{j \in F} \; D[i,j] x_{ij} \tag{18}$$

$$\text{s.t. } \sum_{j \in F} x_{ij} + e_i = s_i + d_i \;, \quad \forall i \in M \tag{19}$$

$$\sum_{i \in M} x_{ij} = p_j \;, \quad \forall j \in F \tag{20}$$

$$\sum_{i \in M} s_i = |V| \tag{21}$$

$$\sum_{i \in M} e_i = |V| \tag{22}$$

$$x_{ij} \geq 0\,, \quad \forall i \in M,\ \forall j \in F \tag{23}$$

Let us $(x^*_{ij})$ be the optimal solution of the IP model. We define $x^*_i$ as the vector composed of the $|F|$ entries of $x^*_{ij}$. To introduce the GCC constraints, we need to define a new variable $J^i_r$ which specifies which forest area will be visited just after unloading request $r$ at mill $i$. The added variables and constraints are thus:

$$J^i_r = F_{S_r},\ \ \forall i \in M, \forall r \in I : M_r = i, \tag{24}$$

$$GCC(x^*_i, F, J^i_r),\ \ \forall i \in M. \tag{25}$$

The starting and finishing woodmills associated to each truck are determined by the IP model, it is possible that a truck begins its day at a woodmill $m$ and finishes it at an another one. For simplicity reasons, we use the linear programming solver (CPLEX) imbedded in OPL to solve the IP model.

### 3.3 Exploring other IP solutions

In some cases, it is possible that the flow structure of the IP optimal solution generates additional waiting time for trucks and forest log-loaders. It is thus interesting to explore other structurally different solutions of the IP model, which can be communicated to the CP model. These solutions could generate less trucks and log-loaders waiting time than the optimal solution without deteriorating too much its value.

To generate new solutions, we perturb the objective function by introducing a penalty term on the value of the maximum flow in the IP solution. The model that we propose will no longer decrease the deadhead distance as the penalty coefficient increases (see Theorem 1). Basically, this means that we can increase the penalty coefficient until we consider that the empty driven distance is sufficiently deteriorated.

The objective function of this IP model is to minimize the deadhead plus penalty term which corresponds to the weighted value of the maximum flow. To present this model we need to define some new parameters and variables.

$c$ : penalty coefficient.
$v$ : the maximum number of empty trucks driving from a woodmill to a forest area,
$P_c$ : the perturbed IP model associated with penalty coefficient $c$,
$X^*$ : a vector composed of the $|F||M|$ entries $x^*_{ij}$,
$(X^*_c, v^*_c)$ : the optimal solution of $P_c$,
$W(X^*_c)$ : the empty driving distance of the optimal solution of $P_c$.

$$\text{Minimize} \sum_{i \in M} \sum_{j \in F} D[i,j] x_{ij} + cv \tag{26}$$

$$\text{s.t.} \sum_{j \in F} x_{ij} + e_i = b_i + d_i \ , \forall i \in M \tag{27}$$

$$\sum_{i \in M} x_{ij} = p_j \ , \forall j \in F \tag{28}$$

$$\sum_{i \in M} b_i = |V| \tag{29}$$

$$\sum_{i \in M} e_i = |V| \tag{30}$$

$$x_{ij} \leq v \qquad \forall i \in M, \ \forall j \in F \tag{31}$$

$$x_{ij}, v \geq 0 \qquad \forall i \in M, \ \forall j \in F \tag{32}$$

**Theorem 1** *: Let $c, c'$ be two integers such that $c < c'$, then $W(X^*_c) \leq W(X^*_{c'})$.*

*Proof.* Assume for the moment that $v^*_c < v^*_{c'}$.
Since $(X^*_{c'}, v^*_{c'})$ is the optimal solution associated with $P_{c'}$, we know that

$$W(X^*_{c'}) + c' v^*_{c'} \leq W(X^*_c) + c' v^*_c. \tag{33}$$

We have assumed that $v^*_c < v^*_{c'}$ and we have $c < c'$, therefore:

$$(c - c') v^*_{c'} < (c - c') v^*_c \tag{34}$$

$$(33) + (34) \Rightarrow W(X^*_{c'}) + c v^*_{c'} < W(X^*_c) + c v^*_c, \tag{35}$$

which is impossible, since $(X^*_c, v^*_c)$ is the optimal solution associated with $P_c$. It follows that

$$v^*_c \geq v^*_{c'}. \tag{36}$$

The fact that $(X^*_c, v^*_c)$ is the optimal solution of $P_c$ implies that

$$W(X^*_c) + c v^*_c \leq W(X^*_{c'}) + c v^*_{c'}. \tag{37}$$

$$(37) - c(36) \Rightarrow W(X^*_c) \leq W(X^*_{c'}) \tag{38}$$

As previously noted, the results of the $P_c$ problem are communicated to the CP model by introducing new global cardinality constraints GCC.

## 4 Experimental Results

In this section we compare the three methods presented previously and evaluate their respective performance on two different case studies. These cases where provided by the Forest Engineering Research Institute of Canada (FPInnovation[1]). The instances[2] come from two large timber companies. The first case involves six forest locations and five woodmills and the average cycle time to transport a load is 5.3 hours. The second case has five forest locations, five woodmills and an average cycle time of 4.7 hours. In both cases, loading and unloading times take around 20 minutes, with loading taking somewhat more time. With problem data discretized in five-minute units, the loading and unloading times of the first case are respectively 20 and 15. For the second case study the average loading time of 21 minutes and an average unloading of 18.5 minutes, have been both approximated to the nearest multiple of 5 minutes which is 20 minutes.The data of these case studies will be available on the web.

We ran several scenarios for each instance, varying the time horizon and amount of logs that needed to be transported (about 10-15 loads are carried every 6 hours). In the first set of experiments (reported in Tables 1 and 2), the number of vehicle varies between 14, 16 and 18 in order to ensure at least three and at most four trips per truck for each scenario. Each scenario was run for exactly 60 minutes using either the basic or the decomposition approach. The value of each solution is presented in terms of unproductive costs. We report (in dollars) the deadhead costs, the waiting cost of trucks queueing to get loaded or unloaded, the waiting cost of log-loaders waiting for a truck to arrive, and finally the total cost of all these activities.

[1] The 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.

[2] Instances are available at www.crt.umontreal.ca/∼louism

| $\lvert V \rvert$ | $\lvert R \rvert$ | deadhead($) | truck($) | log-loaders($) | total($) |
|---|---|---|---|---|---|
| Basic Approach | | | | | |
| 14 | 45 | 5600 | 40 | 2675 | 8315 |
| 16 | 55 | 6907 | 80 | 2125 | 9112 |
| 18 | 70 | 8680 | 1935 | 2442 | 13057 |
| Decomposition Approach | | | | | |
| 14 | 45 | 5512 | 125 | 3633 | 9270 |
| 16 | 55 | 6801 | 250 | 3675 | 10726 |
| 18 | 70 | 8569 | 190 | 5608 | 14367 |

**Table 1. Results of the first case study**

| $\lvert V \rvert$ | $\lvert R \rvert$ | deadhead($) | truck($) | log-loaders($) | total($) |
|---|---|---|---|---|---|
| Basic Approach | | | | | |
| 14 | 45 | 6399 | 95 | 5325 | 11819 |
| 16 | 55 | 7845 | 170 | 5991 | 14006 |
| 18 | 70 | 9800 | 325 | 6092 | 16217 |
| Decomposition Approach | | | | | |
| 14 | 45 | 3202 | 145 | 4983 | 8330 |
| 16 | 55 | 3914 | 25 | 5825 | 9764 |
| 18 | 70 | 4982 | 45 | 8066 | 13093 |

**Table 2. Results of the second case study**

Analyzing the results of Tables 1 and 2, we note that the cost of the unproductive activities increases proportionally with the size of the instances.

For the first case study, it seems that the decomposition could not improve the best solution of the straightforward method, as even a slight reduction of deadheads can significanlty increase the waiting costs of the truck and the loaders. However, in the second case study, the decomposition method generally provides a better overall solution than the straightforward approach. As anticipated, for the larger instances, the log loader waiting time considerably increases as a result of the network structure that is imposed in the scheduling problem through the cardinality constraints.

When we look at the convergence of the two approaches (basic in Table 3 and decomposition in Table 4), we first notice that in almost half of the scenarios the best solutions did not change after thirty minutes of computation time. We also can note that the decomposition approach converges more rapidly than the basic approach (half of the scenarios converged in 10 minutes by using the decomposition approach, whereas all the scenarios using the basic approach continued to be improved after 10 minutes). However, even when the decomposition approach continued to improve the solution after 10 minutes, the relative savings were not very important (less than 1% of the best solution). This behaviour is naturally explained by the fact that the solution space of the decomposed model is considerably smaller due to the added GCC constraints.

In order to improve the decomposition approach, we proposed the perturbed model (see subsection 3.3) that attempts to explore other circulations of trucks that have less impact on the log-loaders scheduling, but still have a very low deadhead cost. The result of this approach demonstrates that the structure imposed by the chosen vehicle circulation can have a significant impact on the objective function. Admitting the fact that the decomposition approach converges quickly, and that the possible benefit after 2 minutes of computational time is not considerable, we propose to vary the penalty coefficient as much as possible while respecting the total computational time. We limit the penalty coefficient to thirty ($c \leq 30$), and for each value of penalty, we fix the computational time to 2 minutes. Thus the total computational time remains 60 minutes.

| $\|V\|$ | $\|R\|$ | 2min | 5min | 10min | 15min | 30min | 60min |
|---|---|---|---|---|---|---|---|
| Basic Approach | | | | | | | |
| 14 | 45 | 8488 | 8432 | 8432 | 8378 | 8315 | 8315 |
| 16 | 55 | 9298 | 9194 | 9152 | 9152 | 9112 | 9112 |
| 18 | 70 | 13182 | 13174 | 13165 | 13157 | 13132 | 13057 |
| Decomposition Approach | | | | | | | |
| 14 | 45 | 9304 | 9304 | 9270 | 9270 | 9270 | 9270 |
| 16 | 55 | 10760 | 10760 | 10726 | 10726 | 10726 | 10726 |
| 18 | 70 | 14434 | 14434 | 14421 | 14421 | 14407 | 14367 |

**Table 3. Results of the first case study for six different computation times**

| $\|V\|$ | $\|R\|$ | 2min | 5min | 10min | 15min | 30min | 60min |
|---|---|---|---|---|---|---|---|
| Basic Approach | | | | | | | |
| 14 | 45 | 12146 | 12129 | 12054 | 11972 | 11972 | 11819 |
| 16 | 55 | 14127 | 14118 | 14084 | 14068 | 14038 | 14006 |
| 18 | 70 | 16251 | 16244 | 16225 | 16217 | 16217 | 16217 |
| Decomposition Approach | | | | | | | |
| 14 | 45 | 8402 | 8402 | 8345 | 8345 | 8345 | 8330 |
| 16 | 55 | 9782 | 9782 | 9774 | 9774 | 9774 | 9764 |
| 18 | 70 | 13098 | 13098 | 13093 | 13093 | 13093 | 13093 |

**Table 4. Results of the second case study for six different computation times**

| $\lvert V\rvert$ | $\lvert R\rvert$ | deadhead($) | truck($) | log-loaders($) | total($) |
|---|---|---|---|---|---|
| the first case study | | | | | |
| 14 | 45 | 5518 | 120 | 2350 | 7988 |
| 16 | 55 | 6860 | 90 | 2483 | 9401 |
| 18 | 70 | 8645 | 835 | 2217 | 11697 |
| the second case study | | | | | |
| 14 | 45 | 3202 | 135 | 4700 | 8037 |
| 16 | 55 | 3914 | 50 | 5750 | 9482 |
| 18 | 70 | 4982 | 50 | 8066 | 13098 |

**Table 5. Decomposition Approach with perturbation** $c \leq 30$

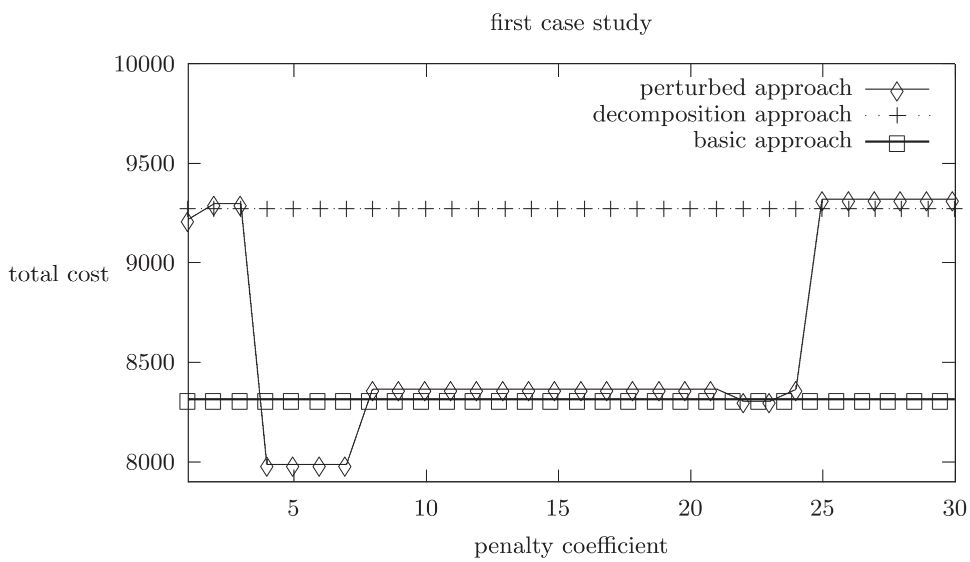


**Fig. 2.** Results of the first scenario of the first case study

Looking at Tables 5 1 and 2, we notice that in five of the six scenarios the perturbed decomposition provides a better or equivalent solution to the straightforward approach. Furthermore, on all our experiments, this later approach could always find its best solutions within 10 minutes of CPU time.

Interestingly, as we see in Figures 2 and 3, the main weakness of the decomposition approach with perturbation is that the perturbed IP model generates

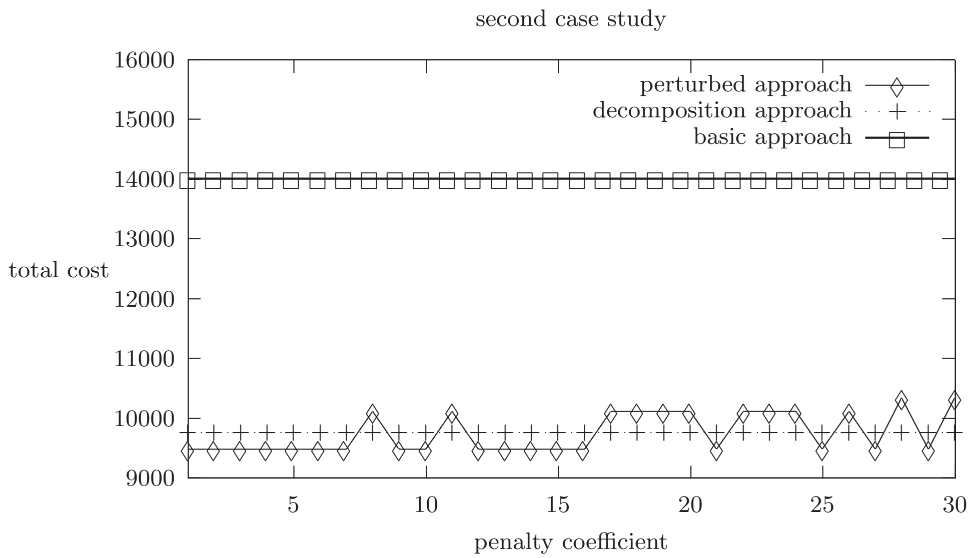


**Fig. 3.** Results of the second scenario of the second case study

the same optimal solution for several penalty coefficients and we thus do not generate enough structurally different truck circulations to communicate to the CP model. Finding a technique to generate truck circulations that are significantly different is thus the subject of future research.

## 5 Conclusion

We have presented a Log-Truck Scheduling Problem with synchronization constraint between the trucks and the log-loaders. To address this problem we proposed a decomposition approach based on Constraint Programming and Integer Programming models, which are combined through the communication of global constraints.

To avoid local minima, a perturbation technique was introduced in the IP model to explore different solutions that can be provided to the CP model.. The whole process was then implemented through a multi-start strategy where we limit the CPU time in order to explore solutions generated with various values of the penalty term.

We believe that if we were able to generate and communicate structurally different optimal solutions of the IP to the CP model, we would be able improve the final result. Other research directions involve solving larger instances with a weekly long horizon and inventory constraints at the mill. These kinds of problems are of great interest to an industry who tries to adopt just-in-time delivery policies.

## References


1. Andersson. G., Flisberg. P., Liden. P., Rönnqvist. M. *RuttOpt- a decision support system for routing of logging trucks*. Can. J. For. Res. 38. 1784-1796 (2008).
2. 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.
3. Baptiste, P., C. LePape, and W. Nuijten. *Constraint-Based Optimization and Approximation for Job-shop Scheduling*. In AAAI-SIGMAN Workshop on Intelligent Manufacturing Systems, pages 5–16, IJCAI'95, Montreal, Canada, 1995.
4. Baptiste, P., C. LePape, and W. Nuijten. *Constraint-Based Scheduling*. Kluwer Academics Publishers, 2001.
5. Fahle, T., M. Sellmann. *Constraint programming based column generation with knapsack subproblems*. In Proceeding of the International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimisation Problems(CP-AI-OR'00), 33–43, 2000.
6. 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.
7. Flisberg. P., B. Liden, M. Rönnqvist. *A hybrid method based on linear programming and tabu search for routing of logging trucks*. Computers and Operations Research ,(In Press).
8. Gronalt, M., and P. Hirsch. *Log-Truck Scheduling with Tabu Search Strategy*. Metaheuristics, Vol 39: 65–88, 2007.
9. Haase. K., G. Desaulniers, and J. Desrosiers. *Simultaneous vehicle and crew scheduling in urban mass transit systems*. Transportation Science, 35(3):286–303, 2001.
10. Hooker, J.N. *A Hybrid Method for the Planning and Scheduling*. Constraints, Vol 10(4):385–401, 2005.
11. Linnainmaa, S., J. Savalo, and O. Jokinen. *EPO: A knowledge based system for wood procurement management*. Paper presented at the 7th Annual Conference on Artifical Intelligence, Montreal, 1995.
12. Milano, M. *Constraint and integer programming*. Kluwer Academic Publishers, 2004.
13. 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.
14. 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.
15. 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.

16. J.-C. Régin. *Generalized Arc Consistency for Global Cardinality Constraint*. In *Proceedings of AAAI/IAAI*, volume 1, pages 209–215. AAAI Press/The MIT Press, 1996.
17. Rönnqvist, M. *Optimization in forestry*. Mathematics Programming, 97:267–284, 2003.
18. 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.
19. Rönnqvist, M., H. Sahlin, and D. Carlsson. Operative planning and dispatching of forestry transportation. Report LiTH-MAT-R-1998-18. Linköping, Sweden.
20. 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.
21. Rousseau, L-M., M. Gendreau, and G. Pesant. *Solving small VRPTWs with constraint programming based column generation*. In Proceeding of the fourth International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimisation Problems (CP-AI-OR'02),333–344, 2002.
22. Sakkout, H.E., and M. Wallace. *Probe backtrack search for minimal perturbation in dynamic scheduling*. Constraints, Vol 5(4):359–388, 2000.
23. Sellmann, M., K., Zervoudakis, P., Stamatopoulos, and T. Fahle. *Crew Assignment via Constraint Programmging: Integrating Column Generation and Heuristic Tree Search*. Annals of Operations Research, 115:207–225, 2002.
24. Simonis, H., P., Charlier, and P. Kay. *Constraint handling in an integrated transportation problem*. IEEE Intelligent Systems, Vol 15(1):26–32, 2000.
25. Van Hentenryck, P. *Constraint Satisfaction in Logic Programming*. MIT Press, 1989.
26. Van Hentenryck, P. *The OPL Optimization Programming Language*. MIT Press, 1999.
27. Van Hentenryck, P., L., Perron, and J F. Puget. *Search and Strategies in OPL*. ACM Transactions on Computational Logic, (1),2:285–320, 2000.
28. 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.
29. Zweben, M., M. Fox. *Intelligent Scheduling*. Morgan Kaufman, 1994.