A Log-Truck Scheduling Model Applied to a Belgian Forest Company

Jean-Sébastien Tancrez 1
Louvain School of Management, Université catholique de Louvain, Mons, Belgium
{js.tancrez@uclouvain.be}

Abstract. Timber transportation is a major activity in the forest industry and leads to challenging operational research problems. In this paper, we study a log-truck scheduling problem that aims at designing the weekly schedule for the transportation of timber from harvest to demand points. In this problem, drivers start and end their working day at home and have a limited working time per day. The quantity available in harvest points is typically several truckloads and full truckloads are thus supposed. To model this problem, we propose an integer program which does not consider the order in which forests are visited by a driver, simplifying its solution. The log-truck scheduling problem we are analyzing has been inspired by the case of a Belgian forest company. The results of our model on this industrial case are given, showing significant improvement compared to the actual schedule used by the company.

Keywords: Timber transportation, log-truck scheduling, optimization model, integer program, case study.

1 Introduction

The forest industry is of significant importance in many countries throughout the world. For example, in Europe, the forest sector’s contribution to the GDP is around 0.8% (2.1% in Northern Europe) [5]. Its activities range from managing the forests and logging trees, to transporting the logs to sawmills or paper mills and producing wood planks or paper sheets. Forest companies often have complex supply chains, and challenging strategic and operational management questions, giving rise to interesting operations research problems [2, 12].

In particular, the timber transportation represents a good share of the costs in the forest industry. The transportation from forests to mills accounts for around 25-40% of the procurement costs [1]. Its scheduling is studied in the scientific literature under the denomination "log-truck scheduling problem" (LTSP, see [1] for a recent survey). This problem aims at designing a minimum cost schedule for the fleet of vehicles (and for their drivers) in advance for a given period (e.g. some days or one week). Of course, the mills demand has to be satisfied, without exceeding the available supply at harvest points. Other constraints may then be added, such as time windows (at the mills or in the forests), driving time limits, or departure from and return to one or multiple vehicle depots. The log-truck scheduling problem is closely related to the pickup and delivery problem [13], but carries some specificities such as [1]: harvest points are visited multiple times as the supply volume is greater than a full truckload; most shipments are full truckloads; a vehicle may visit a supply or a demand point several times in one route; allocations decisions maybe included regarding which forests supplies which mill.

The log-truck scheduling problem has been fairly intensely studied since around 2000. In 2003, Murphy [9] proposed two integer programs to solve two distinct case studies in New Zealand. He supposes full truckloads, a scheduling period of one day, and relies on an arc-based formulation of the vehicle routing problem. The cases are solved using a commercial optimization solver, the first case with 4 forests, 13 mills and 18 trucks, the second case with 5 forests, 7 mills and 12 trucks. To go further, many authors have analyzed more complex versions of the LTSP and proposed solution procedures, often relying on multiple stages approaches, column generation, local search or tabu search. Palmgren et al. [10] study the LTSP with time windows and present results on a case study. They propose a two-phase approach based on
column generation. Columns correspond to feasible routes and branch and price allows to introduce new columns during the procedure. Rix et al. [11] also propose a column generation solution procedure for the LTSP, and apply it to several case studies. They include decisions regarding loader assignment, wood allocation and inventory, and model it as a mixed integer linear program. Gronalt and Hirsch [6] apply the unified tabu search method to the LTSP with time windows. They show the effectiveness of their approach on instances with 560 forests, 3 mills and 10 trucks. Similarly, Flisberg et al. [4] propose a tabu search method, where flows between supply and demand points are defined in a first phase, and the vehicle routing is decided in a second phase. The two-phase approach is applied to a set of industrial cases in Sweden. Derigs et al. [3] tested an alternative metaheuristic, the multilevel neighborhood search, for full truckload log-truck scheduling problems. El Hachemi et al. [8] study a variation of the LTSP where special attention is given to the synchronization of forest loaders and trucks arrivals. Their two-phase solution procedure combines an MIP, local search and constraint programming, and is tested on two industrial cases in Canada. Subsequently, El Hachemi et al. [7] analyzed the weekly LTSP with full truckloads and proposed a two-phase approach based on integer linear programs.

In this paper, we propose a model for a log-truck scheduling problem. This version of the problem has been inspired by an industrial case coming from a forest company in Belgium. It aims at finding the weekly schedule for drivers which have limited working hours every day and park their trucks at their personal home every night. We do not include time windows (for visiting the mills or the forests). The company has around 100 supply points (forests) per week, two demand points (mills) and 18 trucks (or drivers). Our contributions in this paper are in the model we propose and in the solution of the industrial case. To the extent of our knowledge, the proposed formulation has originality in the sense that the order information is not included in the model. The order in which the harvest points are visited by a truck on a given day does not have to be included for the LTSP we are aiming to solve, simplifying its resolution significantly.

The remainder of the paper is organized as follows. In Section 2, we describe our log-truck scheduling problem in details and present our mathematical formulation. Then, in Section 3, we show the results of our case study and discuss the main insights. Finally, we conclude in Section 4.

2 Problem and model

2.1 Problem description

The log-truck scheduling problem we are studying in this paper has been directly inspired by the case of a Belgian forest company. The assumptions were made to comply with the real case, while being able to solve it in reasonable computational time. The LTSP is illustrated in Figure 1, with an example of a driver’s route over one day.

![Figure 1: Problem instance and example of a truck route over one day.](image)
The goal of the LTSP is to design a weekly schedule for drivers to transport timber from harvest points (forests) to demand points (mills). The available timber quantities at forests are given. The quantity available at one forest is often larger than one truckload. A driver must start and end its working day at his home, where he parks the truck during the night. We suppose that a truck is always parked empty during the night and starts a working day by visiting a forest. This assumption is not very constraining, and can anyway be removed by adapting the model (see Section 2.3). The maximum number of working hours per day, e.g. 12 hours, is an important constraint of the problem. We suppose that the harvest points and the mills are always accessible during the working hours, i.e. we do not include time windows.

In our version of the LTSP, we suppose that the destinations of the timber available at a harvest point are known. In other words, there is no allocation decision in our problem. In our industrial case, the two demand points are sawmills that are able to saw timbers of distinct sizes, and the size of the timber available at a harvest point is known when designing the schedule (one harvest point may have timber of different sizes). Moreover, we suppose only full truckload, i.e. the available timber quantities are given as a multiple of one truckload. This is also a reasonable assumption in our industrial case and, in the literature, the LTSP is recognized as mostly full truckload [1].

As destinations are known, a good proportion (around half) of the arcs of a driver’s route, the return trips from forests, are predetermined. This limits the impact of the optimization on the total distance traveled by the drivers. For this reason, and due to the cost structure of our industrial case, the objective of our problem is to minimize the number of working days needed to transport the timber to the mills. As a consequence, the optimization will lead to minimize distances (distances from/to homes and from mills to forests) in order to minimize the total working time, as well as to best fill the working days of the drivers, minimizing periods where they are not busy (similar to a knapsack problem). Note that the working time includes the driving time as well as the loading time at the harvest points and the unloading time at the mills.

2.2 Notations

Before presenting the mathematical formulation of our log-truck scheduling problem, we introduce the notations used in the model. The values given under parentheses are those of the industrial case of the Belgian forest company.

Indices

c ∈ C = \{1 \ldots N_C\} : drivers (N_C = 18).
d ∈ D = \{1 \ldots N_D\} : days (N_D = 5).
m ∈ M = \{1 \ldots N_M\} : mills (N_M = 2).
i ∈ F = \{1 \ldots N_F\} = \bigcup_m F_m : forests, i.e. harvest points (N_F = 114). Note: \(F_m\) is the set of forests with timber with destination mill \(m\). If different timber types are available at one physical location, the harvests points \(i\) are differentiated.

Parameters

\(q_i\) : the timber quantity (in truckloads) to ship from forest \(i\) during the week.
\(t_i\) : time to go from forest \(i\) to the mill demanding the corresponding timber type.
\(t_{mf_{mi}}\) : time to go from mill \(m\) to forest \(i\).
\(th_{fc}\) : time to go from the home of driver \(c\) to forest \(i\).
\(tm_{hc}\) : time to go from mill \(m\) to the home of driver \(c\).
\(tlo\) : time to load a truck in a forest (60 min.).
\(tan\) : time to unload a truck at a mill (45 min.).
T: maximum working time during one day for a driver (720 min.).

M1, M2: large numbers.

Variables

\( x_{cd}^{im} \): number of loadings by driver \( c \) in forest \( i \) coming from mill \( m \), in day \( d \).

\( y^{cd} \): = 1 if driver \( c \) works in day \( d \), 0 otherwise.

\( x_{f_i}^{cd} \): = 1 if driver \( c \) goes first to forest \( i \) in day \( d \) (coming directly from his home, and then going to the corresponding mill), 0 otherwise.

\( x_{m}^{rd} \): = 1 if driver \( c \) leaves mill \( m \) last in day \( d \) (to go back directly to his home), 0 otherwise.

\( z_{cd}^{im} \): = 1 if, in day \( d \), driver \( c \) goes to a forest from mill \( m \) and directly comes back to the same mill \( m \) ("loop" shipment) at least once, 0 otherwise.

2.3 Mathematical formulation

Our log-truck scheduling problem can now be formulated as a linear integer program. As such, this model can be implemented quite easily in a commercial optimization solver.

\[
\begin{align*}
\text{min} & \quad \sum_{c,d} y^{cd} \\
\text{s.t.} & \quad \sum_{m,c,d} x_{cd}^{im} + \sum_{c,d} x_{f_i}^{cd} = q_i & \forall i \\
& \quad \sum_{m} x_{cd}^{im} \cdot (tm_{fi} + ti + tlo + tun) + \sum_{i} [x_{f_i}^{cd} \cdot (th_{fi} + ti + tlo + tun)] + \sum_{m} x_{m}^{rd} \cdot tmh_{mc} \leq T & \forall c,d \\
& \quad \sum_{m} x_{im}^{cd} + \sum_{i} x_{f_i}^{cd} + \sum_{m} x_{m}^{rd} \leq M_1 \cdot y^{cd} & \forall c,d \\
& \quad \sum_{i} x_{f_i}^{cd} = y^{cd} & \forall c,d \\
& \quad \sum_{m} x_{m}^{rd} = y^{cd} & \forall c,d \\
& \quad x_{m}^{rd} + \sum_{i \in F} x_{im}^{cd} = \sum_{i \in F_m} \left[ x_{f_i}^{cd} + \sum_{k \in M} x_{ik}^{cd} \right] & \forall c,d,m \\
& \quad \sum_{i \in F_m} x_{im}^{cd} \leq M_2 \cdot z_{cd}^{im} & \forall c,d,m \\
& \quad z_{cd}^{im} \leq \sum_{i \in F_m} \left[ x_{f_i}^{cd} + \sum_{k \in M/m} x_{ik}^{cd} \right] & \forall c,d,m \\
& \quad x_{im}^{cd}, x_{f_i}^{cd}, x_{m}^{rd}, z_{cd}^{im} \in \mathbb{Z}^+ & \forall i,m,c,d \\
& \quad y^{cd}, x_{f_i}^{cd}, x_{m}^{rd}, z_{cd}^{im} \in \{0,1\} & \forall i,m,c,d
\end{align*}
\]

The objective function (1) minimizes the total number of working days to transport all the available timber. Constraint (2) requires that all the available timber is shipped (to the corresponding mills). Constraint (3) ensures that the total working time of one driver during one day does not exceed the limit \( T \). The working time includes driving, loading and unloading, and accounts for the regular shipments as well as the journeys from home and back home. Constraint (4) ensures that a working day is accounted as soon as one shipment
is carried out that day. Constraint (5) verifies that, when working, a driver has exactly one first shipment from home (directly to a forest). Similarly, constraint (6) verifies that, when working, a driver has exactly one last shipment back home (directly from a mill). Constraint (7) ensures that the number of shipments leaving mill \( m \) is equal to the number of shipments arriving to mill \( m \) (in one day, for one driver). This constraint balances the flow of trucks at the mills. Constraints (8) and (9) are necessary to avoid having balanced shipments at more than one mill while having no shipment linking these mills. In other words, if there are "loop" shipments from and to a given mill (8), the truck has first to arrive to this mill from home or from another mill (9). Constraints (10) and (11) define the variables as integer and binary variables.

The formulation (1-11) gives a base model for our LTSP. In the following, we list several variations that are easy to include and worth mentioning.

- The objective function can easily be adapted to minimize the total working time (and thus a very good proxy for the total distance). For that, the objective function can simply be replaced by the sum over the drivers and the days of the left-hand side term of (3). The two objective functions could even be combined (cost per hour and cost per day). It is likely that these changes in the objective function would not change the results dramatically. Furthermore, with additional data, the objective function could easily be modified to minimize the total distance of the routes, as the variables completely determine the routes traveled by the drivers during the week.

- The model could be straightforwardly extended to allow drivers to come back home with a loaded truck, i.e. through a harvest point. This gives a little more flexibility to design the drivers schedule, but, in our observations, it does not have a significant impact on the final results. Similarly, one could easily include the possibility of journeys between mills, but this also has a negligible impact.

- Constraint (3) forces to respect the maximum working time during one day. Another similar constraint could easily be added to limit the total working time of a driver during one week. The left-hand side term of (3) is simply summed over the days of the week. Rules on the driving time (rather than the working time) could also be added.

- The drivers’ availability, for example holidays, can easily be included by constraining the corresponding variables \( y^{cd} \). Similarly, the model could force a minimum number of working days per week for available drivers.

In the literature, the log-truck scheduling problem is often compared to the pickup and delivery problem [13], and thus considered as an extension of the vehicle routing problem. Accordingly, the order in which the harvest points are visited is most often explicitly considered in mathematical formulations [6, 9] or when generating routes in the column generation [10, 11]. For example, in similar configurations than ours, Murphy [9] includes an index \( l \) to the variables of his integer program to represent the order number of a trip (or load) in a route. Gronalt and Hirsch [6] explicitly include the succession of tasks when defining the variables of their MIP ("\( x_{ijr} = 1 \), if task \( j \) is visited directly after task \( i \) with truck \( r \)"). Our modeling differs from these in that the order information is not included. The schedule of a driver is a list of tasks without specifying the order at which these tasks need to be performed. The order can be decided independently without affecting the length of the routes or the number of working days to transport all the available timber. Of course, for this, the balance of the truck flows need to be verified, and the corresponding constraints (7-9) need to be included in the model. This is possible due to some specificities of our case and of our version of the LTSP, such as the absence of time windows and the full truckload assumption (also true in [9]). It is an interesting feature of our model that makes it simpler to solve, allowing us to find a near-optimal weekly schedule for our case in a short computational time (few minutes).

3 Results and case study

In this section, we apply our model to the industrial case proposed by a Belgian forest company. The case we study aims at scheduling the transportation of logs coming from 114 harvest points, going to 2 sawmills, over 5 days, and with 18 available drivers (or trucks). To illustrate the case, the locations of the
forests, sawmills and drivers’ homes are shown in Figures 2 and 3. The data we got from the Belgian forest company included the driving times between the various points in the supply chain ($t_i$, $tm_{fm}$, $th_{ff}$, $tm_{hm}$) and the list of quantities to harvest in every forest for a week $q_i$ (and the correspondence with the demanding mill). These quantities were given for four weeks. As these were past weeks, we also had access to the actual schedule used by the company for these weeks. This allowed us to compare our results and evaluate the benefits of our approach.

The model has been implemented in a commercial optimization solver, and the industrial case is solved on a laptop (Intel(R) Core(TM) i7 2.60GHz, 8GB RAM). The computational time is limited to 300 seconds (5 minutes). The results are summarized in Table 1, for the four weeks of data available. First, we observe that our model finds near-optimal solutions in a short computational time. The average gap between the best found solution and the lower bound is 3.2%. In practice, it is better as the lower bound could be rounded to the higher integer value, leading to an average gap of 2.3%. In the least favorable case, the proposed solution could only be improved by one or two days only.
Table 1: Overview of the results on the industrial case.

<table>
<thead>
<tr>
<th></th>
<th>Week 1</th>
<th>Week 2</th>
<th>Week 3</th>
<th>Week 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nb. of work. days (best found)</td>
<td>64</td>
<td>53</td>
<td>71</td>
<td>70</td>
</tr>
<tr>
<td>Lower bound on nb. of days (300s.)</td>
<td>62.5</td>
<td>51.2</td>
<td>68.3</td>
<td>67.8</td>
</tr>
<tr>
<td>Optimal gap (in 300 sec.)</td>
<td>2.30%</td>
<td>3.39%</td>
<td>3.77%</td>
<td>3.21%</td>
</tr>
<tr>
<td>Computational time to best sol. (sec.)</td>
<td>10</td>
<td>106</td>
<td>205</td>
<td>212</td>
</tr>
<tr>
<td>Best solution in 30 seconds</td>
<td>64</td>
<td>54</td>
<td>72</td>
<td>n/a</td>
</tr>
<tr>
<td>Real nb. of days used by company</td>
<td>75</td>
<td>60</td>
<td>82</td>
<td>80</td>
</tr>
<tr>
<td>Improvement brought by best sol.</td>
<td>14.7%</td>
<td>11.7%</td>
<td>13.4%</td>
<td>12.5%</td>
</tr>
<tr>
<td>Total nb. of truckloads</td>
<td>190</td>
<td>156</td>
<td>192</td>
<td>200</td>
</tr>
<tr>
<td>Average work. time per day (min.)</td>
<td>703</td>
<td>695</td>
<td>695</td>
<td>703</td>
</tr>
<tr>
<td>Total working time (*12h.)</td>
<td>62.5</td>
<td>51.2</td>
<td>68.5</td>
<td>68.3</td>
</tr>
<tr>
<td>Nb. of forest per driver.day</td>
<td>2.96</td>
<td>2.94</td>
<td>2.70</td>
<td>2.86</td>
</tr>
<tr>
<td>Nb. of used drivers</td>
<td>14</td>
<td>14</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>Nb of days (11h. per day)</td>
<td>71</td>
<td>59</td>
<td>82</td>
<td>79</td>
</tr>
<tr>
<td>Nb of days (13h. per day)</td>
<td>59</td>
<td>49</td>
<td>65</td>
<td>65</td>
</tr>
<tr>
<td>Nb of days (week of 6 days)</td>
<td>64</td>
<td>53</td>
<td>72</td>
<td>72</td>
</tr>
<tr>
<td>Nb of used drivers (week of 6 days)</td>
<td>15</td>
<td>11</td>
<td>16</td>
<td>17</td>
</tr>
</tbody>
</table>

It is also interesting to note that, when compared to the actual schedule that was applied by the company, our model brings a significant improvement by reducing the number of working days, reducing it by 13.1% on average. This improvement may not apply directly if our approach was used by the company due to its assumptions (neglecting the uncertainty in particular), but a good share of the benefits may certainly be reached in reality (for example, in practice the maximum working time per day is not a strict constraint).

Table 1 also shows various metrics regarding the total quantity of timber to collect over the weeks, the working days of drivers, etc. Close to 200 trucks are loaded every week. The average working day of a driver lasts 699 minutes on average. This is quite close to the maximum of 720 minutes, and shows that, unsurprisingly, the optimization leads to fill the working days of drivers as much as possible. This is confirmed when looking at the total working time (as a multiple of 12 hours). We see that it is close to the number of working days used in the best found solution. Table 1 also shows that a truck is loaded around 3 times per day on average. Interestingly, the results also indicate that all 18 available drivers (and trucks) are not used every week. It thus seems that the company could consider buying more timber as it has capacity to transport it.

Finally, Table 1 shows results to illustrate the impact of interesting parameters, i.e. the daily working time limit and the number of working days in a week. Of course, if the days are shorter (longer), significantly more (less) working days and drivers will be needed. However, we also see that the number of working days per week (5 to 6) has a limited impact on total number of working days required to transport the available timber. In fact, with 6 days a week, the solutions tend to deteriorate as the problem becomes more complex and more computational time is needed to reach good solutions. An impact can also be seen on the number of drivers used at least once during the week, as the solver can now schedule 6 working days for some drivers.

4 Conclusion

In this paper, we have analyzed a log-truck scheduling problem inspired by a real case from a Belgian forest company. The specificities of the case have lead to the features of the LTSP we studied, such as: a working day ends and starts at a driver’s home; the daily working time is limited and is an important feature of the problem (similar to a knapsack problem); the cost function is proportional to the number of working days; full truckloads may be supposed; time windows at the harvest and demand points are not included.
The characteristics of the case and the problem allowed us to propose a simple formulation, which is easy to implement and solve. In particular, the model does not include the order in which forests are visited by a driver. While the LTSP is often related in the literature to pickup and delivery problems, and thus to the vehicle routing problem, the order of the nodes in the route do not have to be included in our modeling and truck flow balancing is sufficient.

When applying our approach to the inspiring case, our model shows to be efficient, finding good solutions (average optimality gap of 2.3%) in a reasonable time (few minutes). The results reveal a significant improvement (between 10% and 15%) compared to the actual schedule used by the company. Finally, we feel that the main avenue for future research is to better understand why the specificities of our LTSP allow us to propose a simple modeling for it (no order information), and eventually explore if this modeling can be generalized to other versions of the LTSP. Moreover, as suggested by the company, we could explore ways to deal with randomness in practice (driving time, access to forests).

Acknowledgment

The author is grateful to Daniele Catanzaro (UCLouvain) and Arnaud Willocq for many helpful and interesting discussions during the course of the project.

References