Tour Generation for Log Truck Scheduling

T. F. Robinson
Logging Industry Research Organisation
PO Box 147
Rotorua
New Zealand

Abstract

The Log Truck Scheduling Problem is a special case of the Vehicle Scheduling Problem, with time windows, multiple origins, multiple products, multiple destinations, non-identical vehicles, but at most one load per vehicle at any one time.

A good approach to this problem is to partition it into two parts: tour generation, to schedule a single truck, and tour selection, to direct the tour generation process and select a set of tours that covers the required pickups and deliveries. This paper describes a practical dynamic programming algorithm for log truck tour generation.

1 Introduction

The New Zealand forest industry spends over half a million dollars a day on transporting logs by truck. Even a modest reduction in this cost would save the industry millions of dollars a year and improve its competitiveness in the world market.

The industry has tended to look to truck technology for cost savings, and it seems that each generation of log trucks is lighter, stronger and more versatile than the last. This process of continual improvement shows no signs of stopping, or even slowing; recent innovations include central tyre inflation, on-board computers and GPS [1].

On the other hand, very little has been done in the area of log truck scheduling and despatching. Forest Products Ltd (now CHH Kinleith Region) developed a mechanical despatching aid in the 1960’s [2] (still in use), and another company made an abortive attempt at computerised despatching in the 1980’s, but most New Zealand log truck despatchers have only pencil and paper to help them. Only very recently has even the idea of Computer Aided Despatching gained wide popularity in the industry.

1.1 Definitions

The elementary components of a log truck schedule are:

- load: a collection of logs of a particular length and grade;
- truck: a vehicle capable of carrying one load at a time;
- fleet: the set of available trucks;
forest: a place where log loads are picked up;
customer: a place to which log loads are delivered;
base: a place where trucks are kept overnight;
site: a forest, customer, or base;
trip: a journey made by a truck, loaded or unloaded, between sites;
tour: the sequence of trips (with their times) made by a single truck in a day;
schedule: the set of tours performed by the trucks of a fleet.

Each forest has a known supply of logs of various grades and lengths, and each customer has a known requirement. The scheduler is free to match up supplies and demands as convenient; so long as each customer receives the right log types, it doesn't matter what forest they come from.

Each log truck is capable of carrying only a certain range of lengths, which means that some log loads are incompatible with some log trucks.

Loaded trips are made from forests to customers only; unloaded trips may be base-to-forest, customer-to-forest or customer-to-base. Each truck has exactly one base, and the truck’s tour begins and ends at the base. (The empty tour, having no pickups or deliveries, is also permitted.) Although technically a tour is a sequence of trips, it is more usually handled as a sequence of pickups and deliveries, as in Table 1. A medium sized log truck despatch operation has about twenty trucks, eighty loads, twenty forests and ten customers.

<table>
<thead>
<tr>
<th>Logs</th>
<th>Pick Up From</th>
<th>Deliver To</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1m Kraft</td>
<td>Neptune 43</td>
<td>Technopulp</td>
</tr>
<tr>
<td>12m J grade</td>
<td>Jupiter 29</td>
<td>The Wharf</td>
</tr>
<tr>
<td>7.4m Peeler</td>
<td>Mercury 36</td>
<td>Cyberboard</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Logs</th>
<th>Pick Up At</th>
<th>Deliver At</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:21 am</td>
<td></td>
<td>7:09 am</td>
</tr>
<tr>
<td>8:41 am</td>
<td></td>
<td>10:59 am</td>
</tr>
<tr>
<td>1:44 pm</td>
<td></td>
<td>5:04 pm</td>
</tr>
</tbody>
</table>

1.2 Time and Time Windows

The amount of work a truck can do in a day is constrained in various ways by law and company policy, but in practice this all boils down to a upper limit on tour duration. This limit is either 810 or 840 minutes (14 hours), depending on how scrupulous the forestry company is over rest breaks!

Log loads can only be picked up and delivered during the business hours of the forest and customer respectively. Time windows are not a serious issue in forestry, although there are some customers who prefer to receive their deliveries in the morning. Nevertheless, time windows do exist and their inclusion makes the model more widely applicable.

---

1 In reality, a forest compartment.
2 In applications where each load has a specific customer, this can handled by including the customer name in the log grade.
1.3 Merging Site Activities into Trips

A straight-forward description of log transport by truck would include costs and durations for site activities (loading, unloading, etc.), but with careful formulation, these can be transferred to the trips [3]. (Care is needed because some of the activities at the site depend on where the truck has come from, and some depend on where the truck is going.) This yields the form given below, where activities at the sites are considered to be instantaneous and free.

The reader may be wondering what purpose is served by distorting the natural structure of the problem in this way. The answer is that the problem becomes very much simpler from a mathematical point of view:

1) By concealing site activities in this manner, the number of explicit activities in any tour is halved.
2) When building search trees or networks (see later), each node can be specified by site and time alone. Load information is pushed onto the arcs.
3) The trips defined in this way are completely independent of each other, in that neither their costs nor their durations are influenced by the preceding or following trips. This is useful in some applications, eg. the transportation (stepping stone) formulation in [4].

1.4 The Problem

The problem is to devise a schedule that makes all the required pickups and deliveries at the minimum possible cost. Briefly:

a) The cost of a tour is a flat fee for the truck standing costs, plus the sum of the costs of the component trips.
b) The duration of a tour is the sum of the durations of the component trips.
c) Both the cost and duration of a trip may depend on:
   i) the truck,
   ii) the start and finish sites,
   iii) the load carried, if any.

1.5 Optimal Schedules

The optimal schedule is the schedule which makes the required pickups and deliveries with the least possible cost. There are a number of techniques for finding the optimal schedule, but most of them have a two-phase structure:

a) A method for generating tours;
b) A method for selecting tours to build a schedule.

For example, the problem could be formulated as an integer linear programme with constraints for trucks, pickups and deliveries, and a variable for every possible tour.

2 Tour Generation

Generating a tour is not quite the same as generating a schedule for a single truck, because a tour must pick and choose between the available loads (it cannot carry all of
them). The constraints for tour generation carry over directly from the tour constraints of the master problem, but the objective function requires a bit more thought. The costs of a tour also carry over from the master problem, but if we confine our objective to finding the cheapest possible tour, no logs will get carried at all! In order to get some useful work done, there must be a profit attached to each pickup and delivery. This profit information must come from the tour selection module (e.g. as linear programming shadow prices), since it is not inherent in the scheduling problem.

The methods that immediately spring to mind for this sort of problem are explicit tree search enumeration and network programming.

2.1 Explicit Tree Search Enumeration

For small problems, an explicit tree search enumeration is a viable option. Each tree node has a site (base, forest or customer), an elapsed time, and a cost-so-far. The arcs represent trips (loaded and unloaded), and these arcs can be used to reconstruct the tour by reading back from terminal nodes to the root. The elapsed time at each node is equal to the elapsed time at the preceding node plus the duration of the trip. The cost-so-far at each node is the cost-so-far at the preceding node plus the cost of the trip minus any profit made on the load.

The root node has the truck at its base, with elapsed time and cost-so-far set to zero. Nodes are expanded as follows, in any order (depth first is fine):

1. From the root node, create new nodes for the (unloaded) trips to each forest.
2. Any node which has an elapsed time that exceeds the permitted shift length should be deleted.
3. Any node which has an elapsed time that falls outside the time window for that site should be deleted. (For efficiency reasons, these tests should be made before each new node is created.)
4. From a forest node, create new nodes for trips from that forest to every possible customer with every promising load. A load is promising if it is:
   a) available at the forest
   b) required by the customer
   c) compatible with the truck, and
   d) there is no other promising load which is both quicker and more profitable.
5. From a customer node, create new nodes for (unloaded) trips from that customer to every possible forest, and also for the (unloaded) trip from the customer back to the base. Once a truck reaches the base, its day’s work is over.

If we only want the best tour, there is no need to store the tree explicitly; we can use a set of mutually recursive procedures in the PASCAL sense to make the search. This doesn’t reduce the number of tours however, which is typically over a hundred thousand!

---

3 Since there is only one truck, the standing cost is the same for all tours and can be ignored for the moment.
4 In the PASCAL sense.
2.2 Network Programming

The tour generation problem can be formulated as a network problem, with a node for every site at every time, arcs that represent trips, and trucks as the fluid \[4,5\]. Nodes that have times lying outside the time window for their sites should be omitted. The cost on each arc is the cost of the trip, minus any profit made from the load. The optimal tour is just the shortest path through the network from the base at time zero to the base at any later time. Some arcs have positive cost and some negative, but this is not a problem because the network is acyclic. The solution can easily be found by labelling the nodes in order of increasing time.

The main difficulty with this scheme is that it is necessary to break the day down into discrete time intervals. Working to the nearest minute gives thirty or forty thousand nodes\(^5\).

3 A Pruned Tree Search

To sum up then, both the tree search enumeration and the network programme contain a large number of unnecessary nodes. The tree search suffers from the "combinatorial catastrophe"\(^6\), while the network programme contains nodes that cannot be reached by any tour. (This is especially true for nodes with low elapsed times.) What is needed is a method that has few nodes at the start of the problem, like the tree search, and a manageable number of nodes towards the end, like the network programme. The answer is a pruned tree search. This can also be regarded as a virtual network programme, or a very special kind of branch-and-bound.

The procedure is much the same as the explicit tree search enumeration, with two differences:

1) The tree must be built width first. Nodes are expanded in order of increasing elapsed time.

2) (the supremacy rule) For each site, including the base, a record is kept of the cost of the cheapest tour so far. A node is only expanded if its cost-so-far is less than the cheapest cost recorded for its site.

The requirement that the nodes be expanded in order of increasing elapsed time means that some form of indexing is needed if each next node is to be found efficiently.

Even the pruned tree search can take up a lot of space in the computer, and it is fortunate that the tree can be cut back without losing too much optimality. The supremacy rule must be modified slightly:

2*) ... A node is only expanded if its cost-so-far is considerably less than the cheapest cost recorded for its site.

How much less is considerably less? Fifty cents? Five dollars? The choice of offset is a delicate matter: too small and the programme runs out of space, too large and optimality suffers. Fortunately, the offset can be changed during the course of the search, starting at zero and gradually increasing as memory fills up.

\(^5\) Of course, this can be reduced by a factor of ten just by going to ten minute intervals!

\(^6\) The number of nodes increases exponentially with problem size.
This method works a great deal better than it sounds: picking the best out of 5000 good tours instead of 25,000 good tours has no noticeable impact on the final schedule quality.

4 Computational Experience

LIRO has written an experimental computer programme implementing this method in Turbo Pascal and running on an IBM compatible 486 PC running at 33 MHz. Applied to a sample problem with 74 log loads, 16 pickup points and 9 delivery points, the pruned tree search used 6922 nodes and 36 seconds of computer time. No offset price was needed. For a different truck configuration in the same problem, the tree covered 10392 nodes in 45 seconds; the search only stopped because it ran out of space! Nevertheless, the tour found was shown to be optimal.

This computer programme is still experimental; modularity was more important than speed. The search rate (about 200 nodes per second) is clearly well short of the true potential of the technology.

5 Conclusions

This paper has described the Log Truck Scheduling Problem, a form of Vehicle Scheduling Problem where the tour structure is reasonably straightforward. The greatest simplifying feature is the fact that only one load can be carried at a time.

Three approaches to tour generation have been presented: explicit tree search enumeration, network programming, and a hybrid method incorporating features of both (the pruned tree search). This form of pruned tree search has many applications outside log truck scheduling; indeed, the author has used very similar methods in stock cutting [6].

References