INFOR, Vol. 47, No. 3, August 2009, pp. 215–221
ISSN 0315-5986 | EISSN 1916-0615

A Column Generation Model for Truck Routing in the Chilean Forest Industry

Pablo A. Rey
Escuela de Ingeniería Industrial, Facultad de Ingeniería, Universidad Diego Portales, Santiago, Chile, e-mail: pablo.rey@udp.cl

Juan Andrés Muñoz
División Análisis y Control de Gestión, Gobierno Regional del Maule, Chile, e-mail: jmunoz@coremaule.cl

Andrés Weintraub
Departamento de Ingeniería Industrial, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Santiago, Chile, e-mail: aweintra@dii.uchile.cl

Abstract—This study addresses the problem of scheduling the daily assignment of available trucks for delivery of forest products required at different destinations. The products are logs of various types depending on their point of origin, and are defined in terms of length and diameter. Their destinations include sawmills, pulp mills, and other plants, and ports for export abroad. They are available for pickup and delivery by trucks within a previously defined road network during a working day. The trucks' trip times and load capacities are known. An integer linear programming model is developed for minimizing the costs associated with the daily truck transport operations that satisfy each destination's product demand. The model is based on column generation, each column representing a given truck's trip schedule for a working day feasible for that vehicle. The linear relaxation of the model is solved by dynamically generating columns that are attractive for incorporation and then solving the integer model constructed with all the columns so generated. This approach is then applied to instances whose size and degree of difficulty are similar to those actually encountered in the Chilean forest industry. In every case the linear relaxation optimum is 3% below the integer solution, with execution times low enough to be useful in real-world applications.

Keywords Vehicle routing, column generation, forest management.

1. INTRODUCTION

Forest companies must make daily decisions on how to deliver wood products from points of origin where trees are harvested to destinations such as sawmills, pulp mills and other plants as well as ports for direct shipment abroad. These products consist of logs of various types depending on the zone of origin in the harvest area, and are defined basically by their length and diameter. Daily production at each point of origin, including remainders from the previous day, is a known quantity, as is daily demand at each destination. To ensure proper coordination with downstream operations, daily deliveries are ideally timed to arrive at their destinations at regular intervals.

Trucks typically make various trips per day, picking up logs at origin points and transporting them directly to various destinations. Loading and unloading operations at these points are carried out by cranes and require approximately 20 minutes.

The fundamental decisions regarding log transport relate to what route each truck should take on a given day so as to arrive at delivery destinations on time and at the lowest possible transport cost. The data available are the daily demand for products at destination points, the production and availability of products during the working day at origin points, the previously defined road network connecting origin and destination points, and the costs and times associated with product transport for the different types of truck utilized.

Each type of truck has its particular characteristics, and due to size and load restrictions not all vehicle types can use all of the roads in the network. There are also constraints on the type

Received January 2007; Revision May 2007, Accepted December 2007

of product transported. Some trucks, for example, are specially designed to carry large logs. The problem considered in this study assumes a fixed working day (12 hours), and the specific decisions reflected in the solution are:

- The number of trucks of each type to be employed.
- Truck trip schedules that satisfy operating conditions and demand requirements.
- Daily pickups at origin points and deliveries at destination points.

Each trip is defined in terms of the product pickup at a point of origin, the delivery destination and the departure and arrival times.

Nearly every forest company in Chile and various others in Brazil, Venezuela, Argentina, Uruguay and South Africa have implemented the ASICAM system, which was developed in the mid-1990s by a group of researchers at the Department of Industrial Engineering in the University of Chile led by Dr. Rafael Epstein (Weintraub et al., 1996). This system is based on a simulation framework using heuristics and can solve high-dimensional problems involving up to 250 trucks and 700 trips on a personal computer in a matter of minutes.

Since the first implementation of ASICAM, a number of exact models have been developed for these types of problems that can produce optimal solutions, or at least, solution bounds and feasible suboptimal solutions. Equi et al. (1997), for example, suggest a Lagrangian relaxation algorithm to solve a problem similar to the one presented here below. Among its main characteristics are that it approximates the solution of the Lagrangian dual. Rönnqvist and Ryan (1995) offer an approximate column-generation approach implemented by a New Zealand forest company for supporting real-time decisions.

In Palmgren et al. (2003), a column-generation model is applied to a problem slightly different from the one studied here. In their setup a truck can make various pickups at different origins before making deliveries. Our proposal is similar to the extent that we employ an integer programming model with a restricted set of all possible columns; where we differ, however, is in the method of selecting columns, for whereas Palmgren et al. use a heuristic to generate a pool of available columns, the columns we utilize are generated dynamically for the linear relaxation solution of the integer model. Indeed, our methodology is much closer to the ones devised by Desrosiers et al. (1984) for solving a routing problem with time windows applied to school bus routing and by Desrochers and Soumis (1989) for urban transit crew scheduling.

In the remainder of this paper, Section 2 describes in detail the solved problem and the proposed model while Section 3 reports the results of our computer experiments.

2. MODELING THE TRUCK ROUTING PROBLEM

The truck assignment problem as formulated mathematically exhibits a structure that is amenable to the use of the column-generation method. The basic unit in this type of model consists of columns that are variables or matrix columns in a master problem. Frequently these columns have an intuitive interpretation that allows us to better visualize the problem being modeled. In the present case, a column corresponds to a trip schedule or itinerary for a given truck's working day. The master problem represents the constraints that link the itineraries of the various trucks, which are the satisfaction of demand and the condition that more than one truck cannot be simultaneously loaded at the same network point.

Each truck has a complete itinerary composed of a sequence of visits to origins and destinations with their associated travel times, and a column in the problem indicates the points visited by each truck and the corresponding arrival and departure times.

The formulation of the model begins with a restricted master problem that includes a set of columns which are easy to generate and which guarantee the existence of a feasible solution. Columns are then generated iteratively until the optimal solution of the linear relaxation is found.

Various options exist for column selection. The first one is to generate a set of all possible columns, that is, all possible trips by a given type of truck in one day. In practice, however, this would be very difficult because of the large number of columns to be generated. This was demonstrated by preliminary experiments with problems much smaller than those found in real situations, in which the CPLEX package we used had not found a feasible solution after several hours running time.

An alternative for coping with the large size of the set of all possible columns is to establish some methodology such as a heuristic that selects only a certain number of columns according to user-defined rules, as in Palmgren et al. (2003).

In our case, we chose to solve the linear relaxation of the integer model exactly via dynamic column generation, using the Dantzig-Wolfe methodology for solving large-scale LP problems described in Lasdon (1970). Under this approach, each iteration solves a reduced master problem that includes the initial columns plus all others generated up to that point. Using the dual variables, the reduced costs are calculated for the possible columns not yet generated. If a column is found to have a negative reduced cost it is included in the reduced master problem and the process is reiterated. At each iteration, more than one column can be added to the master problem. Once there are no more columns with negative reduced cost outside the master problem, we have arrived at the optimal solution to its linear relaxation and can thus be sure that the variables corresponding to the columns not generated are all zero at that optimum.

The following points summarize our method of solving the proposed model:

Step 1: Construct a feasible reduced master problem.
Step 2: Solve the linear relaxation of the reduced master problem.

INFOR, Vol. 47, No. 3, August 2009, pp. 215–221
ISSN 0315-5986 | EISSN 1916-0615
DOI 10.3138/infor.47.3.215

Step 3: Solve one or more subproblems to generate negative reduced cost columns.
Step 4: If any negative reduced cost columns are found, add them to the reduced master problem and return to Step 2; otherwise, continue to Step 5.
Step 5: Once all the columns from previous iterations are added as described in Step 4, the integrality of variables condition is restored and the resulting model solved via a branch-and-bound algorithm.

Note that the method does not solve the original integer model exactly. To do so would mean implementing a branch-and-price algorithm as given in Barnhart et al. (1998). However, our computer experiments revealed that the gaps between the linear relaxation solution (which, by contrast, is solved exactly and therefore yields a lower bound for the objective function value) and the integer solutions are around 1% or less with good solution times. This implies that the method is in fact quite suitable for practical applications where models must be solved on a daily basis in limited time periods. For the few cases with bigger gaps, a deep first branch-and-price can produce a better solution in a reasonable time (without exploring the entire tree).

In what follows we first detail the use of a space–time network for defining the columns, and then present the master problem and the column generation subproblems utilized.

2.1 Space–Time Network

An important issue in modeling the problem is how to handle the origin and destination points so that column generation can be used to solve it. As noted in the introduction, average actual truck loading and unloading times for any product are about 20 minutes. Since each origin is defined as a loading crane, the time period cannot be any less than that amount.

Given that in our setting, a truck always loads completely when picking up products at origin points, the mere fact that a truck is making a trip means it will be carrying a full load. A truck schedule consists of a sequence of trips that satisfy certain constraints and/or rules. The rules include the following:

- Loading and unloading schedules must be followed and must concord with product supply and demand, truck loading capacity, quantities available for loading at origin points, and time periods required for travel and loading and unloading operations.
- A truck cannot deliver products it did not previously pick up. If it visits a node, it must later leave it.
- Different product types cannot be mixed on a single trip.

So that the model reflects this reality, the actual origins in the problem are replicated every 20 minutes, meaning that a truck-load of logs can be loaded at that frequency. Thus, we define supply points as [*origin, time*] pairs. In analogous fashion we define demand points as [*destination, time*] pairs that refer to moments when a product is needed at a given destination. Two auxiliary nodes are also defined: Node 0, representing the start of a trip, and Node N + 1, denoting the end of it.

In this auxiliary space–time network each demand point is connected by arcs to each supply point that has products in stock which can be loaded by the type of truck available. Arcs also connect each supply point to each demand point whose requirements can be satisfied by that origin's available products. The definitions of the arcs also take into account travel, loading and unloading times. In precise terms, there is an arc connecting a supply point to a demand point if the product is one of those available at the origin and the destination can be reached within its demand time. Similarly, a demand point is connected to a supply point if the former can be reached from the latter in the corresponding travel time. Arcs also connect Auxiliary Node 0 to all supply points and all demand points to Auxiliary Node N + 1.

The original problem of physical points in a logistical network with associated operation times is thus transformed into a space–time network problem in which each node represents a supply point or a demand point. Feasible daily truck schedules then correspond to paths in the network between auxiliary nodes 0 and N + 1.

The schedules and travel time data are handled implicitly by the auxiliary network. Since the supply points each have a single crane they can only handle one loading operation at a time. Demand points, on the other hand, can unload more than one truck simultaneously as long as they have the necessary cranes. Trucks travel loaded from origins to destinations and return to the origins empty. Figure 1 illustrates the network for our problem.

We now describe in detail the master problem, obtained from the space–time network just described.

2.2 The Master Problem

As was explained in the preceding subsection, the auxiliary network there introduced represents the itinerary of a truck as the path between Node 0 and Node N + 1. Observe that if there are multiple types of trucks, there is an auxiliary network for each type. Every such itinerary is associated with

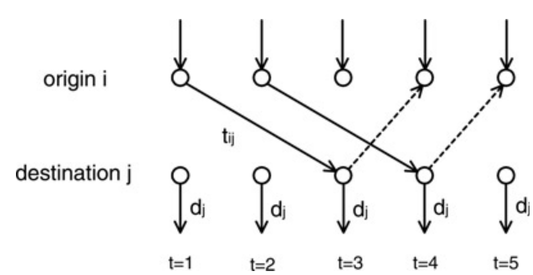

Figure 1. Space–time network for problem: t_{ij} is the travel time between origin i and destination j. Trucks travel loaded from origins to destinations (full arcs) and return to the origins empty (dotted arcs).

DOI 10.3138/infor.47.3.215

a column in the master problem. Once a certain number of itineraries have been generated, it must be decided which of them will be assigned to a given truck (note that not all of them have to be assigned). This is done using binary variables X_{ip} in the master problem that take the following values:

$$X_{ip} = \begin{cases} 1 & \text{if truck type i executes trip schedule } p \\ 0 & \text{otherwise} \end{cases}$$

The master problem [P] then becomes

$$\begin{aligned} [P] \quad & \min \sum_{i=1}^{n} \sum_{p=1}^{m_i} C_{ip} X_{ip} \\ & s.t. \quad \sum_{i=1}^{n} \sum_{p=1}^{m_i} a_{ipo} X_{ip} \leq 1 \qquad o = 1, 2, \ldots . n_O \end{aligned} \tag{2.1}$$

$$\begin{aligned} & \sum_{i=1}^{n} \sum_{p=1}^{m_i} b_{ipd} X_{ip} \geq Dda_d \qquad d = 1, 2, \ldots . n_D \\ & X_{ip} \in \{0, 1\} \qquad \forall\, i, p \end{aligned} \tag{2.2}$$

where

- p schedule ($p = 1 \ldots m_i$),
- n number of different truck types,
- m_i number of schedules for each truck type i ($i = 1, \ldots, n$),
- n_O number of supply points,
- n_D number of demand points,
- Dda_d demand at destination d,
- C_{ip} cost of schedule p for truck type i,
- a_{ipo} 1 if truck type i on schedule p passes through point o and 0 otherwise,
- b_{ipd} amount delivered at point d by truck type i on schedule p.

The b_{ipd} parameter indicating the product amounts delivered at destination points is measured in truckloads. It may take on as many different values as there are different types of trucks capable of delivering logs in a given problem. If in fact there is more than one type, their capacities are normalized and a "base truckload" is defined as the capacity of the smallest available vehicle.

The master problem then consists of minimizing total trip cost, assuming the costs of each trip are known and included explicitly in its objective function. The problem has two constraint blocks: those numbered (2.1), which ensure compliance with supply at supply points and the condition that only one truck can load at a given origin at any time; and those in (2.2), which constrain the model to satisfy customer demand.

Various additional conditions mentioned earlier must also be satisfied in the process of deriving a solution to the problem. They include, among others, satisfying machine capacity at loading points, travel times, and loading and unloading times. These restrictions are treated in the column generation subproblem to be described below.

2.3 The Column Generation Subproblem

The column generation subproblem attempts to find the lowest reduced cost schedule, or one or more negative reduced cost schedules, while satisfying the imposed constraints and/or rules. The subproblem can be stated as a shortest path problem in the auxiliary network defined above. The costs associated with the arcs are defined in such a way that the cost of a route is the reduced cost of the column associated with that route for the basic solution to the linear relaxation of the master problem just solved.

The master problem set out in the preceding paragraphs functions with a dynamic column generator. The columns are added to the master problem as they are generated. The dual variables obtained by solving the linear relaxation of the problem are used to construct the objective function of the subproblem corresponding to the reduced cost of the column with respect to the current basic solution of the master problem. The dual variables are classified into two groups: the π_o, which are associated with the supply constraints (2.1), and the λ_d, corresponding to the demand constraints (2.2).

As already mentioned, our sub-problem consists in finding a shortest path in the auxiliary network. This network is a topologically ordered acyclic network in which the first node is the departure base and the last one is the arrival base. A topological order of a directed graph *G(V,E)* is a labeling of its nodes such that for every one of its arcs (*u*,*v*), *ord(u)* < *ord(v)* (see Cormen et al., 1990, Sec 22.4). In the present case the digraph is obtained by ordering the supply and demand points according to their associated times, with the former preceding the latter for nodes whose time is the same. Within each group of points the order is arbitrary; in our implementation, the nodes were ordered by origin or destination label. Under these conditions the problem of the shortest route between auxiliary nodes 0 and N + 1 can be solved in linear time as a function of the number of arcs in the network even though the cost of an arc may be negative. This is of particular interest in our case, where the cost of an arc in the auxiliary network is the difference between the real cost of the trip defined by that arc less the value of the dual variable corresponding to the master problem constraint associated with the arc destination point. Once the topological order *ord* is defined, the algorithm applied to determine the column of lowest reduced cost is as follows (see Cormen et al., 1990, Sec 24.2):

Step 1: Initialize labels:
For each supply point *o*, define *d[o]* = *cost(0, o)* – *dual[o]*, *pred[o]* = 0;
For each demand point d, define d[d] = ∞, pred[d] = ∞
Step 2: For each network node *u*, following the topological order *ord*:
For each node *v* such that there exists an arc *(u, v)*:
if d[v] > d[u] + cost(u,v) – dual[v]; then
d[v] = d[u] + cost(u,v) – dual[v];
pred[v] = u;

ISSN 0315-5986 | EISSN 1916-0615
DOI 10.3138/infor.47.3.215

where *dual[v]* is the dual variable associated with the node v constraint. Thus, if v is a supply point 0 then $dual[v] = \pi_o$, whereas if it is a demand point, $dual[v] = \lambda_d$. The *d[v]* labels correspond to the upper bounds of the cost of the optimal route between Auxiliary Node 0 and node v for the entire process. At the end of the process each *d[v]* denotes the length of the optimal route from the Auxiliary Node 0 to node v. The *pred[v]* labels refer to the nodes preceding the v nodes on the optimal route from 0 to v. With this information we can readily determine the minimum cost route from node 0 to node N + 1 and, for each demand point v, the minimum cost route from node 0 to node N + 1 for which v is the last destination point visited.

Different policies were tried for generating columns at each iteration. Specifically, a single column for each type of truck and one column ending at each demand point were generated. In the former case, either the column with the most negative reduced costs or the first column found with negative reduced cost could be chosen. On the basis of preliminary experimental results we opted for the latter case, that is, the generation of a column with negative reduced cost ending at each demand point.

3. COMPUTER EXPERIMENTS

To evaluate the performance of our model and the proposed solution we conducted a series of computer experiments with instances of varying sizes and degrees of difficulty. In this section we describe these tests and the results obtained. The generation and solution of the models was implemented in C ++ using the ILOG CPLEX 9.1 library to solve both the linear and the integer problems. We used the barrier method for solving the LPs during the column generation phase. For the final IP we stopped the branch-and-bound once the gap between the best integer solution known and the best bound was less than 0.1%, or after 1,500 CPU seconds. The remaining solver parameters were at their default values. The programs were compiled by GCC 3.3.6. All experimenting was executed on an AMD Athlon 64 PC with 1 GB of RAM running SUSE Linux 10.1.

3.1 Description of Instances

The characteristics of the solved instances are summarized in Table 1 below. Four groups of instances of similar size were generated, with five different networks (origins and destinations) generated for each group and 10 instances with different demands for each network.

Several parameters are represented in Table 1 as indicated by the following column headings: **OR**, the origins in the real problem, each of which correspond to 36 supply points in the associated time-space network; **DR**, the destinations in the real problem; **DP**, the demand points defined by a destination and a time, each point corresponding to a node in the time-space network; **Each DP**, the range of truckloads delivered for each demand point; and **Total**, the number of truckloads moved in the given instance.The characteristics of the trucks used to construct the instances are summarized in Table 2 below. The parameters represented in the table are indicated by the following column headings: **CTL**, the cost per travel hour of a fully loaded truck; **CTE**, the cost per travel hour of an empty truck; **CWT**, the penalty assigned per hour of waiting by a truck at a network node; **CNT**, the cost of adding a truck to the fleet (equals the cost of the arcs emanating from auxiliary node 0); and **LC**, the number of load units a given type of truck can carry (demand at destinations is measured in truck load units). For some supply and demand combinations we generated several instances with different combinations of trucks.

TABLE 1.
Description of instances

	Description					
	Nodes			Truckloads		Travel time (min)
Instances	OR	DR	DP	Each DP	Total	
A	12	6	30	3–7	111–160	40–120
B	12	6	30	6–11	180–160	40–120
C	12	6	30	9–15	251–315	40–120
D	20	6	30	12–20	395–493	40–120

To begin the iterations between the master problem and the subproblem, we must first obtain an initial feasible set of columns to be introduced into the model. The mechanisms for selecting this initial set of columns range from heuristics to simple inspection. After some preliminary experimentation we chose a set consisting of some of the columns for which the truck makes a single loaded trip over the planning horizon plus some other columns for which it makes two such trips. There is a trade-off between the quality of the final integer solutions obtained and the time required to find them, which depends on the number of initial columns. Fewer initial columns usually means shorter solution times but larger integer gaps, while more initial columns yields better integer solutions but potentially much longer solution

TABLE 2.
Transport costs and load capacity of trucks

	Costs per hour				
Truck type	CTL	CTE	CWT	CNT	Load capacity LC
1	1,000	900	90	18,000	1
2	900	800	90	20,400	1
3	1,600	1,400	120	22,800	2
4	1,400	1,200	120	25,200	2

INFOR, Vol. 47, No. 3, August 2009, pp. 215–221
ISSN 0315-5986 | EISSN 1916-0615
DOI 10.3138/infor.47.3.215

times. Also, a balance between one-trip and two-trip columns can generate better integer solutions.

Table 3 exemplifies this tradeoff behavior for two instances and various combinations of initial columns. The first two columns in the table indicate the maximum number of each type of column included for each demand point in the initial master problem. For each instance we report the relative gap between the linear relaxation solution and the final integer solution as well as the total execution time.

The results given in the table for two instances in particular (A1-7 and B4-1) show that there is a relationship between the starting set of columns and final solution quality and time. A possible conclusion from this data might be that gaps improve if the initial master problem contains many columns. The experiments with these and other instances led us to adopt a best strategy for generating the initial set of columns. For each demand point and truck type we included as many one-trip columns satisfying this demand as the number of trucks needed to carry the amount required at that point. We also included twice this number of two-trip columns ending at the same demand point.

TABLE 3.
Solution times and gaps depending on the initial set of columns

Number of initial columns		Instance A1-7		Instance B4-1	
One-trip	two-trip	Gap (%)	Time (sec)	Gap (%)	Time (sec)
20	20	9.0	127	1.2	82
20	50	6.1	117	1.2	139
50	20	3.3	91	1.8	12
50	50	2.2	211	1.4	99
All	0	1.9	267	2.6	42
20	All*	0.0	398	1.7	112
All*	All*	0.0	426	0.2	271

*All columns that could be generated.

As described above, the solution process generates many columns in each iteration. For each truck type and demand point, the column ending there with minimum reduced cost is generated. If this reduced cost is negative, the column is included in the master problem.

3.2 Results Obtained

Tables 4 and 5 below present the results of the experiments conducted. Table 4 reports minimum, maximum and average solution times for each group of instances. The solution times for column generation (linear relaxation solution) and the final integer model solution are given in seconds. Also shown are the gaps between the integer and LP solutions. Table 5 displays the same solution times expressed as percentages of total time, together with the total number of schedules generated and the number used in the final solution.

TABLE 4.
Execution time and gap between solutions

	Column generation (sec)			Integer model (sec)			Total time	Gap (%)		
Instances	Min	Max	Average	Min	Max	Average	Average	Min	Max	Average
A1	2.46	37.95	12.02	0.15	819.72	32.82	44.83	0.00	7.12	0.43
A2	1.90	30.23	9.86	0.18	248.42	9.46	19.32	0.00	5.33	0.19
A3	3.47	42.64	12.52	0.16	466.18	46.09	58.61	0.00	9.58	1.37
A4	3.24	20.38	9.76	0.17	113.69	8.72	18.48	0.00	5.08	0.32
A5	4.44	43.64	12.54	0.23	542.13	42.97	55.52	0.00	11.09	0.86
B1	4.75	33.55	14.60	0.24	1,022.61	66.04	80.65	0.00	2.69	0.29
B2	4.98	36.88	16.59	0.26	382.10	38.39	54.98	0.00	2.22	0.27
B3	4.94	37.93	12.23	0.21	62.17	5.08	17.31	0.00	2.62	0.09
B4	4.39	43.75	14.60	0.24	1,021.64	80.52	95.12	0.00	2.16	0.27
B5	4.95	38.37	14.97	0.25	856.17	49.82	64.79	0.00	2.51	0.21
C1	8.45	27.04	17.46	0.44	2.71	1.16	18.62	0.00	0.08	0.02
C2	6.37	28.36	17.49	0.25	9.86	2.24	19.73	0.00	0.05	0.01
C3	8.68	49.98	19.10	0.37	18.35	4.21	23.30	0.00	0.10	0.01
C4	7.50	50.04	21.66	0.29	527.57	38.75	60.41	0.00	0.10	0.01
C5	7.40	36.76	22.27	0.31	327.38	75.45	97.72	0.00	0.39	0.09
D1	14.91	103.43	39.23	0.49	69.54	12.71	51.98	0.00	0.05	0.01
D2	17.35	64.58	37.53	0.52	1,103.85	145.75	183.28	0.00	0.42	0.05
D3	16.78	71.12	35.26	0.65	315.04	22.54	57.80	0.00	0.01	0.01
D4	16.78	64.87	40.36	0.65	1,301.61	127.38	167.74	0.00	0.51	0.06
D5	17.70	117.74	53.74	0.38	262.39	58.86	112.60	0.00	0.10	0.01

INFOR, Vol. 47, No. 3, August 2009, pp. 215–221
ISSN 0315-5986 | EISSN 1916-0615
DOI 10.3138/infor.47.3.215

TABLE 5.
Distribution of execution time and columns generated

	Column generation			Integer model			Generated columns					
	Time %			Time %			Total			Final solution		
Instances	Min	Max	Average	Min	Max	Average	Min	Max	Average	Min	Max	Average
A	4.21	97.28	82.77	2.72	95.79	17.23	527	4,346	1,515	11	91	56
B	3.18	97.04	80.62	2.96	96.82	19.38	1,080	6,130	2,318	36	152	92
C	7.02	97.71	84.77	2.29	92.98	15.23	1,676	5,800	3,306	72	182	130
D	3.90	97.90	79.38	2.10	96.10	20.62	3,029	9,336	5,205	120	301	192

As is apparent in Table 4, solving the column generation problem at the root node takes less than 2 minutes for all instances. The difficulty of solving each IP problem varies greatly depending on the instance. However, Table 5 shows that on average, just 20% of total time is spent solving the integer model. Note in particular that about 40% of the instances were solved at the root node and the time limit was reached for only 5 instances.

4. CONCLUSIONS

This study addressed the operational problem faced by Chilean forest companies when deciding how to transport their products to customers. The solution proposed was obtained by constructing an integer programming model solved by column generation in which each column represented the daily trips made by a truck between origins and destinations.

A number of ways exist to generate columns or trip schedules using the model constructed. We utilized a dynamic programming column generator to provide new routes at each iteration of the problem. Our model chose few routes at each iteration in an auxiliary network generated using the dual variables of the master problem.

The model was tested for various instances representing real situations faced by forest companies of different sizes. The results showed that the feasible integer solutions were reasonably close to the continuous optimal solution, thus confirming that our model would be usable in real situations. In most test instances, relative gaps of 2% or less were found with execution times that were low enough for practical use in actual applications that require daily trip scheduling.

An interesting task for further exploration would be to combine this model with one for daily harvest planning so as to integrate the two decision-making processes. Attempting such an integration using metaheuristics could possibly lead to significantly suboptimal solutions.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the partial funding for this study provided by FONDECYT Regular Project Nos. 1040520 and 1040727, the Millennium Science Institute "Complex Engineering Systems" and CNPq's PROSUL-Proc. Project No. 490333/04-4. This work was partially developed while the first author was assistant professor at the Departamento de Ingeniería Industrial of the Universidad de Chile.

REFERENCES

Barnhart, C., Johnson, E., Nemhauser, G., Savelsbergh, M., and Vance, P. (1998), "Branch-and-price: column generation for huge integer programs", *Operations Research*, 46: 316–329.

Cormen, T., Leiserson, C., and Rivest, R. (1990), *Introduction to Algorithms*, The MIT Press.

Equi, L., Gallo, G., Marziale, S., and Weintraub, A. (1997), "A combined transportation and scheduling problem", *European Journal of Operational Research*, 97: 94–104.

Desrochers, M. and Soumis, F. (1989), "A column generation approach to the urban transit crew scheduling problem", *Transportation Science*, 23: 1–13.

Desrosiers, J., Soumis, F., and Desrochers, M. (1984), "Routing with time windows by column generation", *Networks*, 14: 545–565.

Lasdon, L. S. (1970), *Optimization Theory for Large Systems*, Macmillan Series in Operations Research.

Palmgren, M., Rönnqvist, M., and Värbrand, P. (2003), "A solution approach for log truck scheduling based on composite pricing and branch and bound", *International Transactions of Operations Research*, 10: 433–448.

Rönnqvist, M. and Ryan, D. (1995), "Solving truck despatch problems in real time", *Proceedings of the 31th Annual Conference of the Operatinal Research Society of New Zealand*, 165–172.

Weintraub, A., Epstein, R., Morales, R., Serón, J., and Traverso, P. (1996), "A truck scheduling system improves efficiency in the forest industries", *Interfaces*, 26: 1–12.

INFOR, Vol. 47, No. 3, August 2009, pp. 215–221
ISSN 0315-5986 | EISSN 1916-0615
DOI 10.3138/infor.47.3.215

Information Systems and
Operational Research

INFOR