Information Sciences 264 (2014) 302–316

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Scheduling a log transport system using simulated annealing

Karunakaran Haridass [a], Jorge Valenzuela [a], Ahmet D. Yucekaya [b,*], Tim McDonald [c]

[a] Department of Industrial & Systems Engineering, 3304 Shelby Center, Auburn University, Auburn, AL 36849-5347, USA
[b] Department of Industrial Engineering, College of Engineering, Cibali Campus, Kadir Has University, Istanbul, Turkey
[c] Department of Biosystems Engineering, Auburn University, 224 Tom E. Corley Building, Auburn, AL 36849-5347, USA

ARTICLE INFO

Article history:
Received 23 November 2012
Received in revised form 29 October 2013
Accepted 8 December 2013
Available online 12 December 2013

Keywords:
Log transport
Simulated annealing
Deterministic simulation
Vehicle routing

ABSTRACT

The log truck scheduling problem under capacity constraints and time window constraints is an NP-hard problem that involves the design of best possible routes for a set of trucks serving multiple loggers and mills. The objective is to minimize the total unloaded miles traveled by the trucks. In this paper, a simulated annealing – a meta-heuristic optimization method – that interacts with a deterministic simulation model of the log transport system, in which the precedence and temporal relations among activities are explicitly accounted for, is proposed. The results obtained by solving a small size problem consisting of four trucks, two mills, three loggers, and four truck trips showed that the best solution could be found in less than two minutes. In addition, the solution method is tested using data provided by a log delivery trucking firm located in Mississippi. The firm operates sixty-eight trucks to deliver loads from twenty-two logging operations to thirteen mill destinations. The routes assigned by a supervisory person are used as a benchmark to compare the manual generated solution to the solution obtained using the proposed method.

1. Introduction

The log truck scheduling problem has been discussed widely in the literatures of transportation and routing management. In forestry, nearly 40% of the costs are involved in road planning and in transportation operations and management. Monetary savings can be realized if some improvement is made to the existing logistics system. In the southern US, trees are harvested using what is termed a 'Tree-length' approach. From there, the tree-length logs are transported to consuming mills and further processed into lumber or paper. The wood products industry in southern US is facing increasing pressure from cheaper wood imports. The need to reduce transportation costs has been felt very acutely among loggers and mill owners. The logging capacity has always been in excess of demand and it is essential for the logger owner to reduce costs in order to survive in business. Usually, a logging company owns a group of trucks and operates them between a single logging site and a set of consuming mills to satisfy the demand. This dedicated trucking capacity is felt to be necessary in order to ensure logs are delivered as quickly as possible, but there is an inherent inefficiency with this approach. Depending on distance to mills and status of unloading facilities, trucking capacity can be either in short supply or severely underutilized. Trucks often spend a great deal of time simply waiting for loading or unloading.

All the upstream and downstream operations depend on the timely pick-up and delivery of wood by the trucks. The inter arrival time of trucks should be as uniform as possible with less variance to ensure smooth inventory level at the pick-up and delivery points. If trucks do not reach the pickup point at the right time, the inventory level at the logger reaches maximum

[*] Corresponding author. Fax: +90 2125334327.
E-mail addresses: valenjo@auburn.edu (J. Valenzuela), ahmety@khas.edu.tr (A.D. Yucekaya), mcdontp@auburn.edu (T. McDonald).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.12.005

Nomenclature

T	number of trucks
M	number of mills
L	number of loggers
N	number of trips
I	denotes truck i (i = 1, ... , T)
m	denotes mill m (m = 1, ... , M)
l	denotes logger l (l = 1, ... , L)
j	denotes trip j (j = 1, ... , N)
L_{ilmj}	equal to one when truck ‘i’ goes from the logger ‘l’ to mill ‘m’ during trip ‘j’
U_{ilmj}	equal to one when truck ‘i’ goes from the mill ‘m’ to logger ‘l’ during trip ‘j’
D_{lm}	distance between logger ‘l’ and mill ‘m’
S_{lm}	load requirement between logger ‘l’ and mill ‘m’

and the logging process gets blocked due to the unavailability of stocking space. Similarly, at the location of mills, there should be sufficient logs available to keep expensive cranes and other Woodyard equipment working at peak capacity.

Ideally, the role of a log transport manager would be to form a demand matrix of loads to be delivered based on data provided by mills and loggers. A distance matrix containing the shortest path between any two locations would then be used to create an efficient transport schedule that met demand for all loads and that minimized the total distance traveled. Unfortunately, this type of system does not exist in practice, resulting in a large percentage of unloaded miles and wasted fuel for loggers, and highly variable utilization of woodyard equipment at consuming mills. An optimization system that effectively allocated and scheduled loads to be delivered would result in greater transport efficiency in wood procurement and lower overall costs for the forest products industry.

The truck routing problem is a complex combinatorial problem and few approaches to solve the problem have been proposed in literature. There are many heuristic approaches like tabu search, column generation, branch and bound heuristics to solve the general vehicle routing problem where the destinations are visited only once as in Audy et al. [1], Benjamin and Beasley [3], Drexl [9], El Hachemi et al. [10,11]. But in the case of the log truck scheduling problem multiple visits to a single mill or logger are required to meet the demand. Since this involves repeated visits to the customer sites, it cannot be easily solved using existing heuristic approaches.

The majority of work done previously discusses solving the vehicle-routing problem where the destinations are visited only once rather than a log-truck scheduling problem. A solution methodology is needed to solve the log-truck scheduling problem which is a special case of vehicle-routing problem where multiple visits to a single mill or logger are required to meet the demand. Gerdessen [13] has made an attempt to solve the vehicle-routing problem with trailers. The problem is solved in two phases. Construction heuristics are first used to obtain a feasible solution and then improvement heuristics are used to obtain an improved solution from the initial solution.

Heuristics and metaheuristic algorithms are common in literature. In [25] simulated annealing and immune system is combined to solve the optimization problems that are taken from the literature. In Yildiz [26,27] Particle Swarm Optimization is used to solve the design problems. In Yildiz and Durgun [28] and Yildiz [29] Cuckoo search is used to solve the design problems and parameter selection problems. In Yildiz [30], a comparison analysis for the population-based optimization algorithms is given. Chao [7] models a truck and trailer routing problem and uses the tabu search heuristic to solve the problem. The customers are assigned to the trucks by relaxing the integrality constraint and obtaining a linear initial solution. The three types of routes namely pure vehicle route, pure truck route and complete vehicle route are constructed in the next step using the cheapest insertion heuristic. Sumichrast and Markham [21] have compared the solutions obtained by a heuristic algorithm with that of a lower bound integer programming solution from a commercial solver. The problem size used in the paper was small in terms of demand met for the day and in the number of trucks used to meet the demand.

Thangiah et al. [22] have developed a heuristic algorithm for the vehicle routing problem with backhauls and time windows. In this paper, the earliest and latest times were defined which indicate the lower bound and upper bound at the customer's site. The loading time, unloading time and waiting time were exclusively accounted for when calculating the total route time of the trucks. Czech and Czarnas [8] have used parallel simulated annealing algorithm to solve a vehicle routing problem with time window without any backhauling. The objective function minimized the number of vehicles used as well as the total distance traveled. The truck could only serve customers whose total demand is less than the total capacity of the truck in one trip.

Audy et al. [1] propose a literature review that includes planning methods and decision support systems for vehicle routing of forest products. They summarize many related vehicle routing literature that are focused on forest transportation. Rix et al. [18] propose a column-generation method to solve the multi-period log-truck scheduling problem. The objective is to find the quantity of wood that needs to be transported that will minimize total cost. The problem is modeled as a mixed integer linear program and solved using column generation in which log-truck routes are the columns. El Hachemi et al. [10] develop a hybrid approach to solve the log-truck scheduling problem using constraint programming and mathematical programming. The problem consists of scheduling the transportation of logs between forest and woodmills as well as routing the fleet of vehicles. El Hachemi et al. [11] develop a heuristic method to solve the log-truck scheduling problem which includes constraints such as pick-up and delivery, multiple products, inventory, multiple supply points and multiple demand points. They develop a decomposition approach in which in first step they determine the destinations of the truckloads and in the second step they schedule the routes.

Azi et al. [2] propose a branch-and-price approach for the vehicle routing problem with time windows and multiple use of vehicles. Column generation and linear programming are used to solve the problem. Another research on vehicle routing that includes time windows, driver rest period and multiple facilities is developed in Benjamin and Beasley [3]. A metaheuristics is proposed to find a solution to the problem. Drexl [9] presents a literature survey on the multiple synchronization constraints which include researches that are recently developed to solve problems with the similar structure.

Ronnqvist [19] describes the scope for optimization in various stages of forestry beginning with harvesting, bulking, crew scheduling, road building, fleet management, truck scheduling and ending with production planning. Solution methods such as dynamic programming, linear programming methods, branch and bound methods, heuristics and column generation were used for the optimization. Bent and Van Hentenryck [4] have modeled the dynamic vehicle routing with stochastic service times and customer locations using the Multiple Plan Approach and Multiple Scenario Approach. The Multiple Plan Approach's main idea is to generate and maintain all possible sets of routing plans along with the main routing plan. Random customer requests are ranked according to a pre-defined priority. The requests are then checked against the available plans. If they match one of the available plans is accepted; otherwise it is rejected. Weintraub et al. [24] discusses the need to centralize ownership of the trucks in order to schedule the trips efficiently. The paper explains that the trucks should start and end the day from a place which is very near to the driver's home. The paper mainly aims at meeting the demand and did not take into consideration any optimization or heuristic approach to reduce the unloaded miles traveled. Palmgren et al. [16] proposed a solution approach based on column generation and pseudo branch and price for the log-truck scheduling problem. The solution approach starts with generating an initial set of feasible routes followed by pricing using dual variables in order to reduce the cost of the routes. Hirsch and Gronalt [12] have used four different tabu search approaches and a few post optimization heuristics to solve the timber transport vehicle routing problem. One standard tabu search approach searches a large neighborhood whereas the tabu search with limited neighborhood approach searches a very limited neighborhood, and the third approach is a combination of the above two approaches. Bent and Van Hentenryck [5] discuss the importance of solving vehicle routing problems with time windows and capacity constraints in a two stage hybrid manner. It was found that when minimizing the dual objective of travel cost and the number of routes, the algorithm succeeded only in minimizing the travel cost. Osman [20] has investigated three different algorithms namely descent, hybrid simulated annealing with tabu search and tabu search for their performance in solving vehicle routing problem with capacity and distance constraints. Tabu search was the best of all algorithms used but it required a lot of storage space for the selection of the alternative solution. Valenzuela et al. [23] have designed a transportation scheduling system for silvi-cultural projects considering project due dates, precedence relationship and transportation costs. They considered five different classes of resources. The solution approach consists of two stages starting with simulating the network and then improving the solution using a search heuristics.

In this paper, a simulated annealing algorithm that interacts with a deterministic simulation model is developed. The simulated annealing algorithm consists of two stages. In the first stage, an initial solution is made feasible by meeting all demands for the day. The solution is then improved using transfer, exchange, interchange and stop operators. Improvement is continued until the user-specified stopping temperature of the simulated annealing process is reached.

The paper has been organized as follows. In Section 2, we describe the problem. In Section 3, we develop the algorithms for the proposed approach. In Section 4, we test the solution approach and compare the results with a manually generated solution. In Section 5, we give the conclusions.

2. Problem description

In a pooled transport system, all loggers and mills are brought under the centralized leadership of a single scheduler. The scheduler determines the movement of the trucks to meet the demand on any given day. The network consists of a hub, a set of loggers and mills distributed near the hub. The mills require logs for further processing and they represent the demand points.

The loggers and mills are the customers for the trucks. The trucks visit the loggers and the mills by traveling the shortest possible distances and delivering the loads as fast as possible. The driver takes the truck at the beginning of the shift, delivers the required number of loads to the specified places, and leaves the truck at the hub once the shift is over. The routing plan consists of a set of routes assigned to all of the trucks. In truth, the routes driven by trucks in a typical log delivery system are rather ad hoc in nature, based more on expediency and luck than any planning. It is our contention that a better approach would be to assign each truck an optimal route to deliver the required loads. The routes should be designed in such a way that the truck travels a minimum of unloaded miles and at the same time the routes meet the demand requirement at the

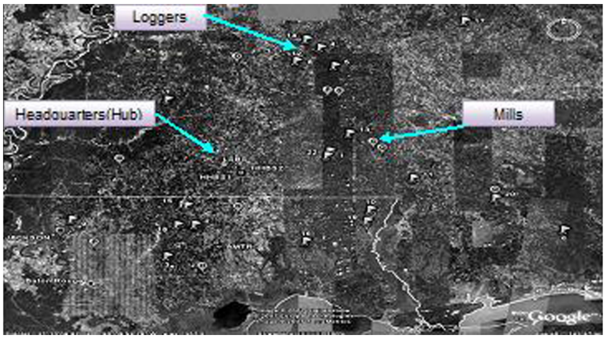


Fig. 1. The distribution of loggers, mills and the headquarter.

end of the day. The normal duration of a shift is eight to ten hours. The truck drivers and the operators at the logger and mill locations are required to work within the same time window. This is specifically done to overcome the labor costs associated with overtime work. Fig. 1 shows the distribution of the loggers, mills and the main hub. The drivers and the operators start working together until end of the shift without any overtime. The delivery schedule enforces all operators to finish their work on time so that trucks can finish their route within given time window.

In the optimization scheme, the locations of the loggers and mills are assumed to be vertices in a two dimensional Euclidean space. A linear programming model for the problem is provided in Appendix A. The time taken to travel the distance between a logger and a mill is proportional to the distance between that particular logger and the mill. The speed of the truck is limited to 50 miles/h. All the trucks have to start from the hub and return to the hub at the end of the day. There is no parking space near the customer's site. For simplicity in routing and management of the trucks, the trucks end the day at the hub. Any truck can serve any customer and a customer can be served by more than one truck on any day depending on the load required for that customer. All trucks are assumed to have an identical capacity. The trucks always travel between the customer locations as fully loaded or empty. These constraints are used as the operating company requires driver to perform under these constraints.

The hub is neither a logger nor a mill. It is just a parking place for all the trucks. All loggers and mills are located at a particular distance from the hub. The distances between loggers and mills are fixed and are input to the problem. Also, the distances between the hub and the customer locations are known. On a particular day, the loads transported between the loggers and mills are fixed.

The trucks should be utilized to the maximum extent possible. Thus, the routes are designed to reduce the number of trucks used. Fewer trucks with high utilization are more desirable than more trucks with low utilization. Also, the total distance traveled by all trucks is minimized. This happens because the objective of the optimization is to reduce the total unloaded miles (the total loaded miles is fixed). The routing plan consists of assigning a predecessor and a successor to each customer location in the truck's route.

3. Simulated annealing approach

Simulated annealing (SA) is a meta-heuristic algorithm used to solve complex large problems having a large solution space and producing results close to the global optimum value in a short period of time. The algorithm is inspired by nature and is derived from statistical mechanics. The SA algorithm was first created by Metropolis et al. in 1953 to generalize the Monte Carlo method to determine the equations of state and also to determine frozen states of n-body systems in the field of metallurgy. In the later part of the 20th century, the concept of SA was put forward independently by Kirkpatrick et al. [14] and also by Cerny [6] as a means to solve large-scale optimization problems. A detailed explanation of the method can be found in Reeves [17].

The proposed S.A. algorithm consists of the following steps.

STEP 1: An initial solution is generated using construction heuristics and the initial solution is accepted as the current solution and the overall best solution. The algorithm begins by fixing a temperature T = Initial temperature.

STEP 2: Using a set of predefined search operators, the current solution's neighborhood is searched. A new solution is identified.

STEP 3: The new solution is input to a deterministic simulator to obtain a suite of performance measures that are used in computing a fitness function. Depending on the value of the fitness function, the new solution can be adopted as the best solution.

STEP 4: The current solution could be replaced by the new solution with a probability depending on the current 'annealing' temperature and the difference in fitness function value between the current solution and the new solution.

STEP 5: The value of the ‘annealing’ temperature is decremented a small step. The value of the temperature is checked to determine whether it has reached its final value. If not, then the steps from 2 to 5 are repeated.

STEP 6: When the final temperature is reached, it is assumed the best possible solution has been obtained.

The proposed SA algorithm consists of two main heuristics. An initial solution is generated using a set of construction heuristics during the initial phase of the simulated annealing process, and then altered using another set of improvement heuristics applied during the neighborhood search. The construction heuristics involve randomly setting the routes of the trucks. Initially, there are two kinds of trucks: loaded and unloaded. The loaded trucks always start the day by delivering their load to a mill, and unloaded trucks start the day by visiting a logger. Once a truck visits a logger, its route should alternate between a delivery point (mill) and another logger. The truck’s last trip always ends at the hub since all the trucks are assumed to return to the hub at the end of the day.

Each trial solution is input to a simulator to generate performance metrics that include number of unmet loads, number of over-met loads, unloaded miles traveled by the trucks, time spent waiting for service at logger and mill locations, and the number of trucks not completing their routes within the operating time window. A fitness function is then calculated by allocating a penalty to each of the performance measures.

$$\begin{aligned}\text{Fitness function}(F) = {} & \text{Travel Penalty} * \text{unloaded miles} + \text{Penalty} * \text{unmet loads} \\ & + \text{Waiting Penalty} * \text{total waiting time} + \text{Excess Penalty} * \text{Over met loads} \\ & + \text{Hub Penalty} * (\text{Trucks that have not completed the trips})\end{aligned}$$

$$\text{Fitness function} : F = K_1 * UM + K_2 * UL + K_3 * OL + K_4 * WT + K_5 * NT$$

The improvement heuristics play a major role in finding a better solution. We will use decrease route operator when there are too many trips scheduled than demanded. Increase route operator will be used when there are unmet loads as a result of unassigned route trips. A truck needs to finish its delivery within defined time window. Stop truck operator is applied when there are one or more trucks that have not returned to the hub within a given operating window. The transfer route operator is used to switch one or more trips from one truck’s route to another truck’s route. The interchange operator, on the other hand implements an exchange of individual trips between trucks. They consist of a set of operators used to perturb a current solution such that a new, feasible solution is generated. The operators are applied collectively or individually based on the value of the performance metrics of the current solution. Care is always taken to ensure the final trip for each truck is always to the hub.

The following are the operators used to improve the current solution.

3.1. Decrease route operator

The decrease route operator is used whenever there are over-met loads. Over-met loads are those not required to meet the day’s delivery schedule and are the result of scheduling too many trips for the trucks. Over-met loads always make a given solution infeasible. In applying the operator, a truck and a trip are randomly selected (from among the set of deliveries that are in excess of that required) and the loggers and mills in that selected trip are deleted. Fig. 2 illustrates an example. It shows the truck’s route before and after applying the decrease operator. Note that logger 4 and mill 5 have been chosen randomly and removed from the truck’s route. The truck’s route has been reduced from 3 trips to 2. This reduces the length of the routes, thereby reducing the truck’s assignments for the day. While using the operator, care should be taken to ensure that the sequence of the route is not altered; i.e., if the truck goes to a logger, it should go to the mill as the next destination and vice versa.

3.2. Increase route operator

The increase route operator assigns more loads to be delivered by a selected truck. This operator is used whenever there are unmet loads. Unmet loads are those in the daily schedule of deliveries for which no truck route has been assigned. In

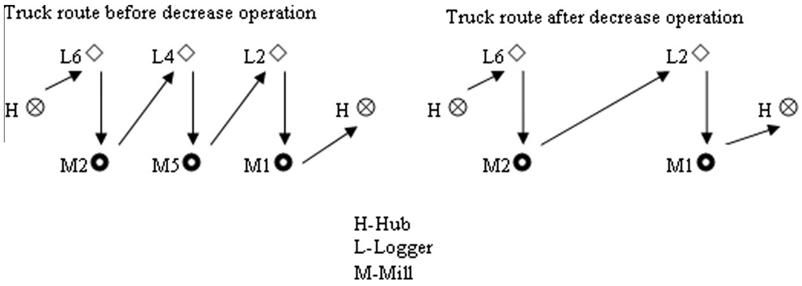

Truck route before decrease operation
Truck route after decrease operation
L6
L4
L2
H
M2
M5
M1
H
L6
L2
H
M2
M1
H
H-Hub
L-Logger
M-Mill

Fig. 2. Decrease operator being applied to cut short the route length.

applying the operator, a truck and trip combination are selected randomly and the unmet delivery is inserted into the truck's schedule before the selected trip. Fig. 3 shows an example. It shows the truck's route before and after applying the increase operator. There were only 2 trips before applying the increase operator and the route length is increased to 3 as the logger 5 and mill 7 pair is inserted. To be eligible for application of this operator, a truck should currently be delivering fewer loads than the maximum number of trips allowed. Again the process of increasing the route is different for a loaded truck and unloaded truck, and the sequence of the route is respected. The effect of the increase-route operation is opposite that of the decrease-route operation and is executed depending on the number of unmet over-served loads in the solution.

3.3. Stop truck operator

Even though all the loads are perfectly met for a given delivery schedule, the time required for a truck to complete its specific route could be very high. There are legal limits imposed on the number of hours a driver may work consecutively, and operations in the woods and at mills typically do not extend into evening hours. The stop truck operator is therefore applied whenever there are one or more trucks that have not returned to the hub within a given operating window, normally 8 h. The stop truck operator is applied to a randomly selected truck and trip combination, and results in all subsequent trips being deleted. Fig. 4 shows an example truck route before and after applying the stop operation. The last trip, consisting of logger 9 and mill 6, is removed from the truck's route. The difference between the decrease operation and the stop operation is that the decrease operation picks any trip in the truck's route and eliminates it, whereas the stop operation tries to eliminate one or more trips towards the end. By doing this, we are trying to stop the truck after a particular number of trips, and sending it back to the hub. The solution may often become infeasible, but it may be again made feasible using the Increase operator or the decrease operator.

3.4. Transfer route operator

The transfer route operator is used to switch one or more trips from one truck's route to another truck's route, or to another part of the same truck's route. The two trucks and trips are randomly selected. The logger and the mill from the first truck's route are inserted into the other truck's route after the randomly selected trip. Fig. 5 shows an example illustrating the application of the transfer operator in which the last trip between logger 6 and mill 2 is transferred from truck 1's route to the last trip in truck 2's route. Before the transfer, truck 1 had 4 trips in its route whereas truck 2 had only 2 trips. By transferring the trip, the two trucks have been given an equal work load. The transfer route is one of the more complicated operators since it can be used to transfer trips within the same truck and also from one truck to another route irrespective of the condition of the truck. By applying this operator, there is a chance that the truck might take the shortest route in deliv-

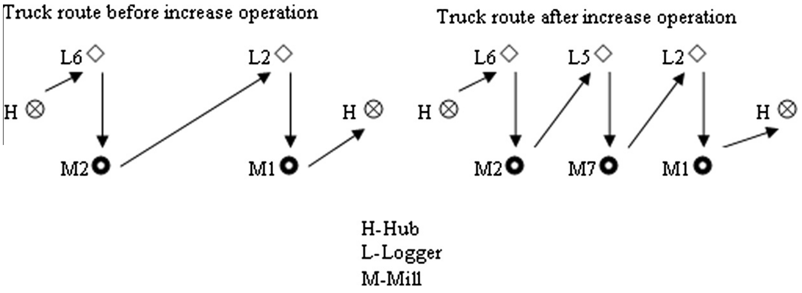

Truck route before increase operation
L6
L2
H
H
M2
M1
Truck route after increase operation
L6
L5
L2
H
H
M2
M7
M1
H-Hub
L-Logger
M-Mill

Fig. 3. Increase operator being applied to increase the route length.

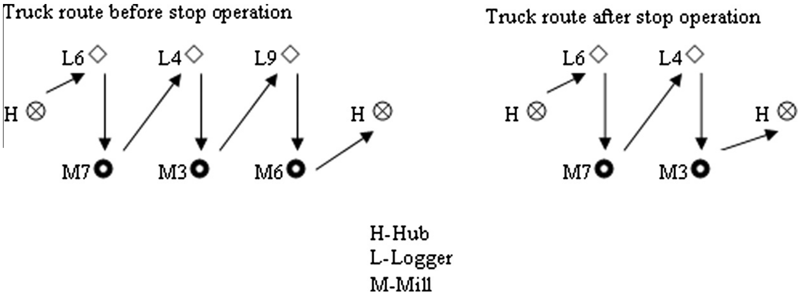

Truck route before stop operation
L6
L4
L9
H
H
M7
M3
M6
Truck route after stop operation
L6
L4
H
H
M7
M3
H-Hub
L-Logger
M-Mill

Fig. 4. Stop operator being applied to stop the truck.

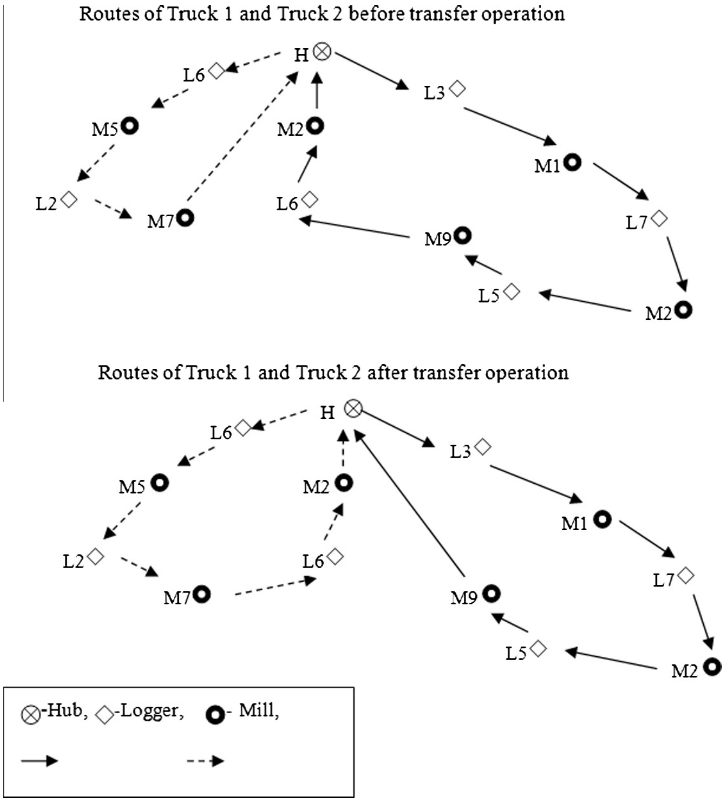

Routes of Truck 1 and Truck 2 before transfer operation
H
L6
L3
M5
M2
M1
L2
M7
L6
L7
M9
L5
M2
Routes of Truck 1 and Truck 2 after transfer operation
H
L6
L3
M5
M2
M1
L2
L6
L7
M7
M9
L5
M2
⊗-Hub, ◇-Logger, ●- Mill,

Fig. 5. Transfer operator being applied to trucks' routes.

ering the loads. The feasibility of the solution is not affected since unmet or over-met loads are not created, although the truck receiving the transferred trip may sometimes take a longer time to complete its route.

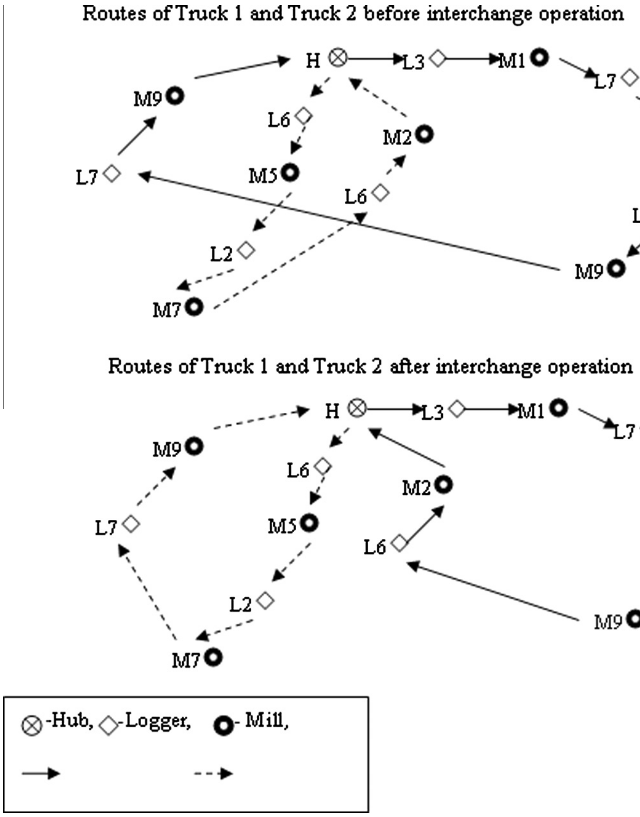
3.5. Interchange operator

The final improvement heuristic, the interchange operator, implements an exchange of individual trips between trucks. Similar to the transfer operator, the interchange operator can be used to switch trips either between two distinct trucks, or to switch order of deliveries within the same truck. The operator can, in some instances, have the effect of reducing the total unloaded miles as well as the total waiting time and at the same time do not impact feasibility of the solution. While interchanging the trips, the first mill of loaded trucks is left undisturbed since that destination cannot be changed. Fig. 6 shows the routes of trucks 1 and 2 before and after an interchange operation. Logger 6 and Mill 2 of truck 1 are interchanged with logger 7 and mill 9 of truck 2.

3.6. Fitness function

The value of the fitness function gives a measure of the quality of a solution, the lower the fitness value the better the solution is thought to be. The above operators when applied to a current solution generate a new solution. The set of routes of the new solution is then given as an input to the simulator along with the description of the problem. The simulator emulates the movement of the trucks and outputs all the performance metrics such as total unloaded miles, total waiting time, total time duration to complete the trips. These performance metrics are used to calculate the value of the fitness function for that solution. If F_1 denotes the fitness value of a current solution, F_2 the fitness value of the new solution, and $\delta = F_1 - F_2$ the difference in the fitness value, then the best solution is updated if $\delta > 0$. The new solution may replace the current solution with a probability that depends on the current temperature and the difference in the fitness value (δ). This condition is given by:

$$\mathrm{Random}(0,1) < \frac{T+\delta}{T}$$


Routes of Truck 1 and Truck 2 before interchange operation

Routes of Truck 1 and Truck 2 after interchange operation

Fig. 6. Interchange operator being applied to trucks' routes.

Random(0, 1) represents a random number between 0 and 1 whereas T is the temperature. If the above condition is not satisfied, then the current solution is maintained, otherwise the current solution is replaced by the new solution. At each temperature level, the neighborhood of the current solution is searched for a fixed number of iterations to find a new solution. The annealing temperature is then decremented by multiplying it by a factor β, which is fixed and remains constant throughout the process. The above process is repeated until the temperature reaches the final temperature value. At the end of the annealing process, the best solution found is printed out. It contains the best set of routes that the trucks should take in order to deliver the loads with fewest unloaded miles, and lowest waiting and total time to complete the delivery schedule. In order to clarify the solution process, the pseudo code of the algorithm is provided in Fig. 7.

4. Numerical results

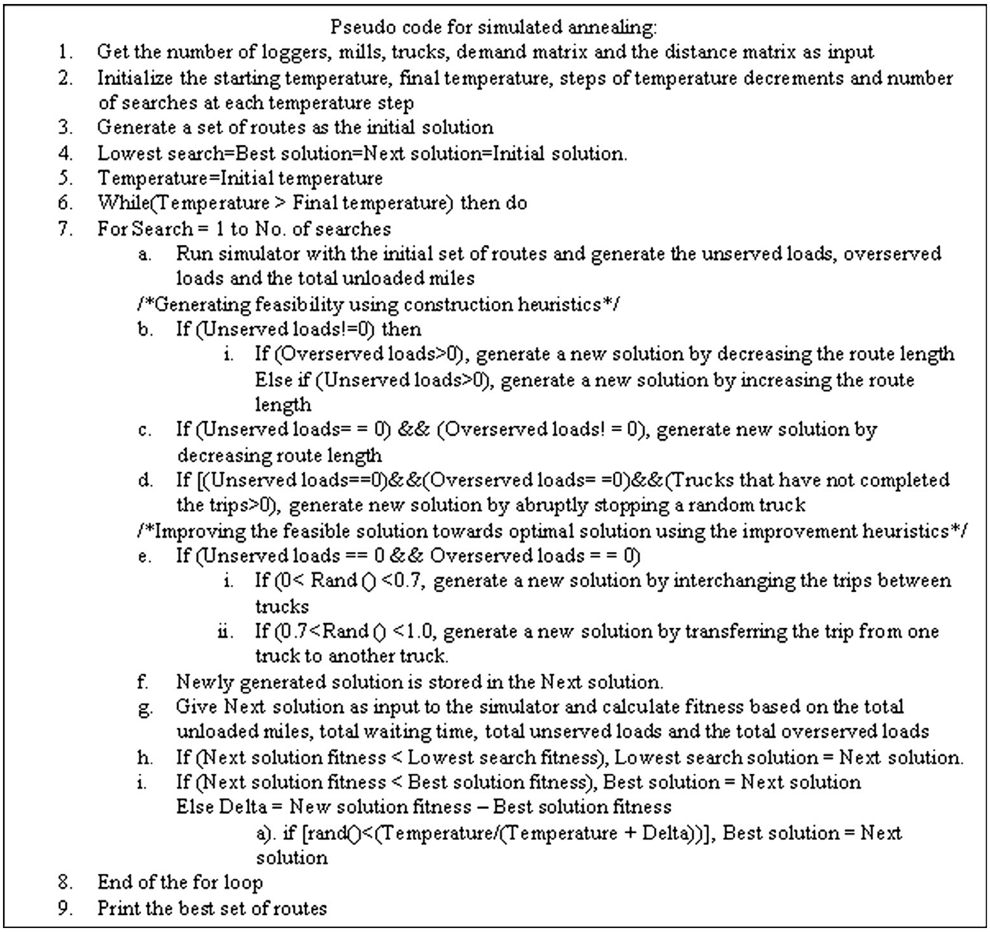
4.1. Solving a test problem

At first, a small problem is solved and compared to the optimal solution obtained using an integer programming model as described in McDonald and Valenzuela [15]. The small problem consists of 4 trucks, 2 mills, 3 loggers, and a maximum of 4 trips per truck. The distances between the mills and the loggers as well as the demands are given in Tables 1 and 2 respectively.

In this problem, all the trucks are initially unloaded and the objective is to minimize the total unloaded miles traveled by the trucks. The optimal solution of the integer programming model is given in Table 3. These routes have a total of 399 unloaded miles.

The above problem is also solved using the proposed approach. After trying different combinations of parameter values, it was decided to set the following values for the parameters of the simulated annealing process: initial temperature = 100, number of searches = 2000, decrementing factor β = 0.9, final temperature = 0.1. For the penalty factors, the following values were used: K_1 = $1/unloaded mile, K_2 = $10,000/unmet load, K_3 = $10,000/overserved load, K_4 = $ 0.01/h of waiting time, K_5 = $1000/ truck not returning to the hub before 8 h. Table 4 shows the best routes found by the SA algorithm.

Although the actual routes are different, the values of the objective functions are the same for the two models. The total unloaded miles found using the SA algorithm is 399 miles also. This indicates that the problem has more than one optimal



```
Pseudo code for simulated annealing:
1. Get the number of loggers, mills, trucks, demand matrix and the distance matrix as input
2. Initialize the starting temperature, final temperature, steps of temperature decrements and number
   of searches at each temperature step
3. Generate a set of routes as the initial solution
4. Lowest search=Best solution=Next solution=Initial solution.
5. Temperature=Initial temperature
6. While(Temperature > Final temperature) then do
7. For Search = 1 to No. of searches
     a. Run simulator with the initial set of routes and generate the unserved loads, overserved
        loads and the total unloaded miles
     /*Generating feasibility using construction heuristics*/
     b. If (Unserved loads!=0) then
          i. If (Overserved loads>0), generate a new solution by decreasing the route length
             Else if (Unserved loads>0), generate a new solution by increasing the route
             length
     c. If (Unserved loads= = 0) && (Overserved loads! = 0), generate new solution by
        decreasing route length
     d. If [(Unserved loads==0)&&(Overserved loads= =0)&&(Trucks that have not completed
        the trips>0), generate new solution by abruptly stopping a random truck
     /*Improving the feasible solution towards optimal solution using the improvement heuristics*/
     e. If (Unserved loads == 0 && Overserved loads = = 0)
          i. If (0< Rand () <0.7, generate a new solution by interchanging the trips between
             trucks
          ii. If (0.7<Rand () <1.0, generate a new solution by transferring the trip from one
             truck to another truck.
     f. Newly generated solution is stored in the Next solution.
     g. Give Next solution as input to the simulator and calculate fitness based on the total
        unloaded miles, total waiting time, total unserved loads and the total overserved loads
     h. If (Next solution fitness < Lowest search fitness), Lowest search solution = Next solution.
     i. If (Next solution fitness < Best solution fitness), Best solution = Next solution
        Else Delta = New solution fitness – Best solution fitness
             a). if [rand()<(Temperature/(Temperature + Delta))], Best solution = Next
             solution
8. End of the for loop
9. Print the best set of routes
```

Fig. 7. Pseudo code for simulated annealing.

Table 1
Distances between mills and loggers (in miles).

	Logger 1	Logger 2	Logger 3
Mill 1	70	65	45
Mill 2	67	76	51

Table 2
Demand of mills from loggers (in number of truck loads).

	Logger 1	Logger 2	Logger 3
Mill 1	1	2	2
Mill 2	2	3	1

solution. Both solutions were obtained in less than 5 s. However, in the case of large problems, the integer programming approach could take hours which is prohibitive for daily operations.

4.2. Solving a real-world problem

The real-world problem consisted of 68 trucks, 13 mills, 22 loggers, and a maximum of 7 trips per truck. The data were gathered from a log delivery trucking firm located in Mississippi, USA. The distances between mills and loggers as well as delivery demands are tabulated in Appendix A. At the start of the simulation period there were 10 trucks with loads remaining from the previous day and 58 unloaded.

A delivery schedule generated manually by an employee of the trucking firm was used as a performance benchmark. This schedule was input to the simulator to obtain reference metrics. Fig. 8 shows the finishing time distribution of the trucks for the reference schedule. The mean finishing time was 7.21 h with standard deviation of 3.61 h. It was observed that the last truck completed its delivery schedule after 18 h of operation. There were about 34 trucks that took more than 8 h to deliver their scheduled loads, and 7 trucks were not used.

The same fitness function parameters used in solving the small problem, along with the demand and distances in Appendix A, were input to the SA program to obtain an improved set of routes. The delivery schedule output from the SA algorithm was used as input to the simulator and the resulting performance metrics were compared against those of the manually-generated schedule. For the reference schedule, the average number of trips per truck required to deliver the loads was found to be seven to complete the schedule within an 8 h time window. For the SA-generated schedule, the average number of trips was reduced to five. Fig. 9 compares the ending time of trucks of both methods. From the figure it was evident that all the trucks finished deliveries within 8 h, with a mean finishing time of 5.76 h and a standard deviation of 2.28 h. There were 28 trucks completing deliveries within 6 h and 5 trucks were not used at all. In Table 5 several performance metrics of both solutions are tabulated and compared.

Table 5 shows that the total distance traveled by the trucks was reduced from 20,630 miles to 17,127 miles, a 17% improvement. Since the total distance traveled was reduced, the trucks were able to meet the schedule in less than the 8 h limit imposed on operating time. Had this improved schedule been implemented, it could have saved overtime pay for the trucking firm, as well as shortened the working day for the loggers and mill owners.

Total loaded distance driven in the manual system amounted to 10,028 miles, and in the SA-generated schedule was improved by 5.17% to 9509 miles. There is almost no means for improvement in loaded miles since fixed loads are to be delivered between source logger and destination mill and the trucks follow the same route in delivering the loads. The small difference in improvement of loaded miles was due to the fact there was a difference in the number of trucks ending the day loaded at the hub. Many fewer trucks in the SA-generated schedule returned to the hub loaded than in the manual system, which resulted in a small improvement in the loaded miles percentage.

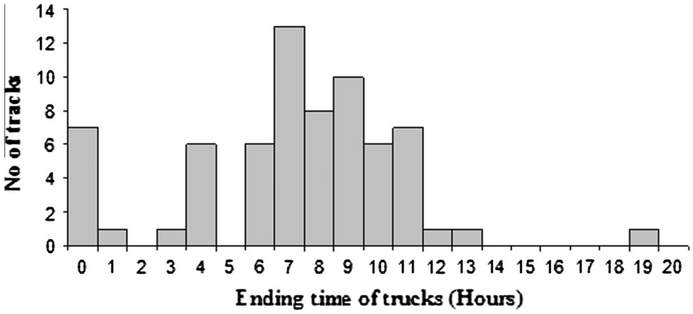


Fig. 8. Finishing time of trucks in the manual routing system.

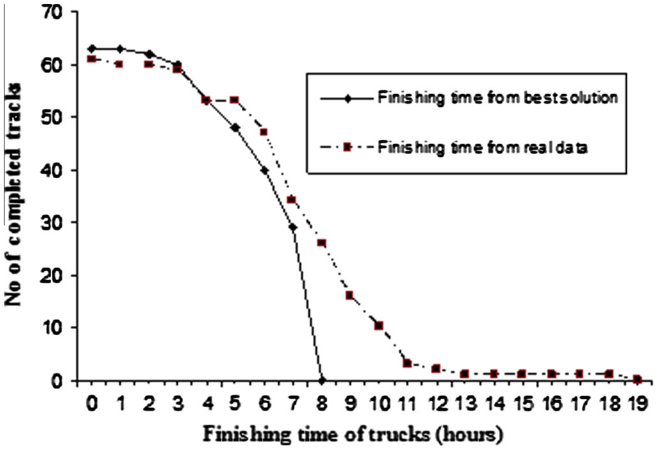


Fig. 9. Finishing time of trucks for manual system and proposed approach.

Table 3
Solution using integer programming approach.

Truck no.	Logger	Mill	Logger	Mill	Logger	Mill	Logger	Mill
Truck 1	2	2	1	2	Hub	Hub	Hub	Hub
Truck 2	2	1	3	1	3	2	Hub	Hub
Truck 3	2	1	2	2	1	2	Hub	Hub
Truck 4	1	1	3	1	2	2	Hub	Hub

Table 4
Solution using SA approach.

Truck no.	Logger	Mill	Logger	Mill	Logger	Mill	Logger	Mill
Truck 1	2	1	2	2	1	2	Hub	Hub
Truck 2	1	1	1	2	2	2	Hub	Hub
Truck 3	2	2	1	2	Hub	Hub	Hub	Hub
Truck 4	2	1	3	1	3	1	Hub	Hub

Table 5
Improvement in performance metrics.

	Manual system	SA approach	Improvement (%)
Total Miles (TM)	20,630	17,127	17
Loaded Miles (LM)	10,028	9509	5.17
Unloaded miles	10,602	7618	28.14
% Loaded Miles (LM/TM)	48.6	55.54	14.28
% Unloaded miles	51.39	44.48	13.72
Total waiting time (h)	21	13.88	33.9

Table 6
Average waiting time at logger locations.

Logger no.	Total number of loads	Average waiting time (min)
1	6	10.00
2	7	1.37
3	11	0.00
4	14	10.71
5	5	0.00
6	3	0.00
7	7	4.29
8	9	4.00
9	5	0.00
10	12	0.50
11	2	0.00
12	18	9.00
13	13	3.97
14	2	0.00
15	1	0.00
16	13	6.92
17	3	0.00
18	6	0.00
19	15	0.00
20	14	4.54
21	3	2.00
22	3	6.00

A truck is said to record unloaded miles when it is traveling from a mill to the logger. There is ample scope for improving the unloaded miles since the truck can always go to the logger nearest to the previous mill visited, and in so doing reduce the total unloaded distance traveled. The improvement heuristic plays an important role in improving the solution by reducing these unloaded miles. The solution generated for this example problem reduced total unloaded miles from 10,602 to 7618,

Table 7
Average waiting time at mill locations.

Mill no.	Total number of loads	Average waiting time (min)
1	15	2.40
2	38	3.36
3	38	8.80
4	4	0.48
5	23	1.36
6	6	0.32
7	39	6.80
8	0	0.00
9	5	0.40
10	2	0.00
11	1	0.00
12	1	0.00
13	0	0.00

an improvement of 28.14%. It increased loaded miles as a fraction of those driven from 49% to 56%, an improvement of 14%. By reducing the total unloaded miles traveled by the trucks, fuel cost is saved and moreover the time wasted in traveling unnecessary unloaded miles is reduced. The improvement heuristics also affected idle time of trucks while waiting for service at loggers and mills. The estimated total waiting time of all trucks in the SA-generated delivery schedule was reduced from 21 h to 13.88 h, an improvement on a per-truck basis of 34%.

Another important factor in optimizing the overall log procurement system is to smooth the flow of material over time between loggers and mills, which tends to reduce waiting times in loading and unloading queues. The SA-generated delivery schedule seemed to provide this benefit when compared to the manual version. The average waiting time of trucks at each logger location is given in Table 6.

On average, the trucks servicing logger 4 had to wait longer (10.7 min) than at other pickup locations. Logger 4 had 14 loads to deliver in the simulated day. The next highest waiting time was for logger 1, which required 10 min waiting time per load, but only had to supply 6 throughout the day. These apparent differences were a result of multiple trucks arriving at a single logger at the same time. Because of the penalty imposed, the improvement heuristics used in this optimization scheme tended to reduce this waiting time and should therefore improve the efficiency of log delivery over manual dispatch.

The average waiting time at each mill location is shown in Table 7. Unlike the waiting times at logger locations, the waiting time at mills are proportional to the number of loads that are to be carried to the mills. The mills 3 and 7 have higher waiting times of 8.80 min and 6.80 min on an average since both mills require a higher number of loads to be delivered to them (38 and 39 loads respectively). At the same time, the trucks arrive at equal intervals to the mill 2 and hence we have a lower waiting time of 3.36 min on an average.

5. Conclusion

In this paper, we described a solution algorithm to solve a log truck scheduling problem with capacity and time window constraints. A SA algorithm that interacts with a deterministic simulation model of the log transport system was developed to solve the problem. The solution time was less than 2 min and the solution outperformed a solution currently generated manually by a supervisory person. The proposed approach was implemented using the C programming language and solved using a computer with Intel Pentium 4 processor having 2.40 GHz of processing speed and 1.5 GB RAM.

The developed algorithm was tested against a small problem with known optimum solution obtained by solving an integer programming model. The performance parameter of total unloaded miles obtained by using the SA approach was found to be exactly the same as that obtained by the integer programming approach even though the solutions were different. The same SA algorithm without any modification was used to solve the practical case where there were comparatively more trucks, loggers and mills. It was found that the unloaded miles were reduced by 28% and the number of trucks needed to serve the loads was also reduced by 17%. There is an absolute necessity for timely deliveries of truck loads which are expected to increase in the future. Hence, the SA algorithm developed in this paper can meet the requirements of the current forest supply chain.

Appendix A

The linear programming of the problem

$$\text{Min Z} = \sum_{i=1}^{T}\sum_{l=1}^{L}\sum_{m=1}^{M}\sum_{j=1}^{N} U_{ilmj} D_{lm} \tag{1}$$

$$\sum_{l=1}^{T}\sum_{m=1}^{M} L_{ilmj} \leqslant 1 \quad \text{for all i} = 1\ldots T \text{ and } j = 1\ldots N \tag{2}$$

$$\sum_{l=1}^{L}\sum_{m=1}^{M} U_{ilmj} \leqslant 1 \quad \text{for all } i = 1\ldots T \text{ and } j = 1\ldots N \tag{3}$$

$$\sum_{k=1}^{L} U_{imkj} \leqslant \sum_{k=1}^{L} L_{ikmj} \quad \text{for all } i = 1\ldots T,\ \ l = 1\ldots L,\ \ m = 1\ldots M \text{ and } j = 1\ldots N \tag{4}$$

$$\sum_{k=1}^{M} L_{ilk(j+1)} \leqslant \sum_{k=1}^{M} U_{iklj} \quad \text{for all } i = 1\ldots T,\ \ l = 1\ldots L \text{ and } j = 1\ldots (N-1) \tag{5}$$

$$\sum_{i=1}^{T}\sum_{j=1}^{N} L_{imlj} = S_{lm} \quad \text{for all } l = 1\ldots L,\ \ m = 1\ldots M \tag{6}$$

$$\sum_{l=1}^{L}\sum_{m=1}^{M}\sum_{j=1}^{N} (L_{ilmj} + U_{ilmj}) d_{lm} \leqslant 400 \quad \text{for all } i = 1\ldots T \tag{7}$$

$$L_{ilmj} \text{ and } U_{imlj} \text{ are the binary variables } (0,1) \tag{8}$$

The main aim of the objective function is to reduce the total unloaded miles traveled by the truck. There are basically six constraints which define the problem space and should not be violated in solving the problem. On any single trip, the truck can travel to the logger only once. The truck can travel from a logger to any mill independent of the mill where it is coming from. The truck can travel from a mill to any logger independent of the logger where it is coming from. The number of loads transported by all the trucks between a logger '*l*' and mill '*m*' must be equal to the load requirement between a logger '*l*' and mill '*m*' during the optimization period.

Distance matrix between loggers and mills (in miles).

Log	Mill													
	1	2	3	4	5	6	7	8	9	10	11	12	13	Hub
1	58	40	42	19	34	48	96	78	71	65	17	110	33	41
2	15	35	62	83	90	56	45	143	82	39	81	41	88	33
3	35	54	74	102	109	67	18	138	102	52	100	36	107	52
4	96	74	90	57	30	98	127	108	28	97	62	141	31	76
5	94	71	78	45	17	84	125	100	33	94	50	139	19	73
6	100	127	63	85	109	61	135	30	168	153	83	180	112	128
7	79	56	110	97	60	116	109	156	29	42	96	88	58	58
8	89	65	85	56	30	93	119	109	20	89	60	133	26	67
9	55	69	102	116	123	96	77	190	116	40	115	21	121	67
10	42	51	7	36	66	13	72	54	99	84	35	97	69	49
11	76	60	46	19	49	39	128	27	103	86	13	131	52	61
12	88	65	85	56	30	93	119	110	20	89	60	133	26	68
13	75	59	45	12	19	51	115	76	63	85	17	130	22	60
14	98	75	97	63	42	135	129	110	30	98	68	143	38	77
15	19	32	66	80	87	59	41	102	80	30	79	43	85	30
16	28	52	28	56	73	31	47	67	99	70	55	75	73	50
17	167	118	129	101	67	128	198	101	99	167	98	212	69	146
18	20	40	67	87	94	61	35	148	87	38	86	36	92	38
19	39	63	2	35	65	8	70	48	110	81	34	95	68	61
20	84	108	46	66	88	45	112	0	147	134	40	165	91	84
21	17	30	63	78	85	57	39	100	77	30	76	45	83	28
22	53	37	45	20	34	46	93	78	80	63	18	108	33	38
Hub	15	2	54	50	60	62	56	109	53	34	48	70	58	0

Demand between loggers and mills (number of loads).

Log	Mill												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	0	3	0	2	0	0	0	0	0	0	0	0
2	4	0	0	0	0	0	3	0	0	0	0	0	0
3	0	9	0	0	0	0	2	0	0	0	0	0	0
4	0	4	9	0	0	1	0	0	0	0	0	0	0
5	0	1	3	1	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	3	0	0	0	0	0	0
7	0	7	0	0	0	0	0	0	0	0	0	0	0
8	0	2	0	0	3	0	0	0	4	0	0	0	0
9	0	4	0	0	0	0	0	0	0	0	0	1	0
10	0	4	2	0	0	0	6	0	0	0	0	0	0
11	0	0	2	0	0	0	0	0	0	0	0	0	0
12	2	0	0	3	9	0	4	0	0	0	0	0	0
13	0	4	2	0	8	0	0	0	0	0	0	0	0
14	0	1	0	0	0	0	0	0	1	0	0	0	0
15	0	0	0	0	0	0	1	0	0	0	0	0	0
16	1	0	4	0	0	3	5	0	0	0	0	0	0
17	0	1	0	0	2	0	0	0	0	0	0	0	0
18	6	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	9	0	0	2	4	0	0	0	0	0	0
20	0	0	4	0	0	0	10	0	0	0	0	0	0
21	0	1	0	0	0	0	2	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	2	1	0	0

References

[1] J.-F. Audy, S. D'Amours, M. Rönnqvist, Planning Methods and Decision Support Systems in Vehicle Routing Problems for Timber Transportation : A Review, Montréal, CIRRELT, 2012. <https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2012-38.pdf>.

[2] N. Azi, M. Gendreau, J.-Y. Potvin, An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles, Eur. J. Oper. Res. 202 (3) (2010) 756–763.

[3] A.M. Benjamin, J.E. Beasley, Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities, Comput. Oper. Res. 37 (12) (2010) 2270–2280.

[4] R. Bent, P. Van Hentenryck, Dynamic vehicle routing with stochastic requests, in: International Joint Conference on Artificial Intelligence, Acapulco, Morgan Kaufmann Publishers, Mexico, 2003. August 9–15.

[5] R. Bent, P. Van Hentenryck, A two-stage hybrid local search for the vehicle routing problem with time windows, Transport. Sci. 38 (4) (2004) 515–530.

[6] V. Cerny, A thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm, J. Optimiz. Theory Appl. 45 (1985) 41–51.

[7] I.M. Chao, A tabu search method for the truck and trailer routing problem, Comput. Oper. Res. 29 (2002) 33–51.

[8] Z.J. Czech, P. Czarnas, Parallel simulated annealing for the vehicle routing problem with time windows, in: 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, Canary Islands, Spain, EUROMICRO-PDP, 376, January 09–11, 2001.

[9] M. Drexl, Synchronization in vehicle routing—a survey of VRPs with multiple synchronization constraints, Transport. Sci. (463) (2012) 1–58.

[10] N. El Hachemi, M. Gendreau, L.-M. Rousseau, A hybrid constraint programming approach to the log-truck scheduling problem, Anna. Oper. Res. 184 (1) (2011) 163–178.

[11] N. El Hachemi, M. Gendreau, L.-M. Rousseau, A heuristic to solve the synchronized log-truck scheduling problem, Comput. Oper. Res. 40 (3) (2013) 666–673.

[12] P. Hirsch, M. Gronalt, Tabu search based solution methods for scheduling log-trucks, Oper. Res./Comput. Sci. Interf. Ser. 39 (II) (2007) 65–88.

[13] J.C. Gerdessen, Theory and methodology vehicle routing problem with trailers, Eur. J. Oper. Res. 93 (1996) 135–147.

[14] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Sci. New Ser. 220 (4598) (1983) 671–680.

[15] T. Mcdonald, J. Valenzuela, Optimizing log transport in the US South, in: Proceedings of the International Information Technology, 2002, Helsinki, Finland.

[16] M. Palmgren, M. Ronnqvist, P. Varbrand, A near-exact method for solving the log-truck scheduling problem, Int. Trans. Oper. Res. 11 (2004) 447–464.

[17] C.R. Reeves, Modern Heuristic Techniques for Combinatorial Problems, McGraw-Hill, 1995.

[18] G. Rix, L.M. Rousseau, G. Pesant, Solving a multi-period log-truck scheduling problem with column generation, in: 34th Council on Engineering Annual Meeting. Québec City, Québec, June 2011.

[19] M. Ronnqvist, Optimization in forestry, Math. Program. Ser. B 97 (1–2) (2003) 267–284.

[20] I.H. Osman, Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem, Ann. Oper. Res. 41 (1993) 421–451.

[21] R.T. Sumichrast, I.S. Markham, A heuristic and lower bound for a multi-depot routing problem, Comput. Oper. Res. 22 (10) (1995) 1047–1056.

[22] S.R. Thangiah, J.Y. Potvin, T. Sun, Heuristic approaches to vehicle routing with backhauls and time windows, Comput. Oper. Res. 23 (11) (1996) 1043–1057.

[23] J.F. Valenzuela, H.H. Balci, T. McDonald, A transportation-scheduling system for managing silvicultural projects, Int. J. For. Eng. 16 (1) (2005) 65–75.

[24] A. Weintraub, R. Epstein, R. Morales, J. Seron, P. Traverso, A truck scheduling system improves efficiency in the forest industries, Inst. Oper. Res. Manage. Sci. Interf. 26 (4) (1996) 1–12.

[25] A.R. Yıldız, Hybrid immune-simulated annealing algorithm for optimal design and manufacturing, Int. J. Mater. Prod. Technol. 34 (2009) 217–226.

[26] A.R. Yıldız, A new hybrid particle swarm optimization approach for structural design optimization in automotive industry, J. Automob. Eng. 226 (D10) (2012) 1340–1351.

[27] A.R. Yıldız, A novel particle swarm optimization approach for product design and manufacturing, Int. J. Adv. Manuf. Technol. 40 (2009) 617–628.

[28] A.R. Yıldız, I. Durgun, Structural design optimization of vehicle components using Cuckoo search algorithm, Mater. Test. 54 (2012) 185–188.

[29] A.R. Yıldız, Cuckoo search algorithm for the selection of optimal machining parameters in milling operations, Int. J. Adv. Manuf. Technol. 64 (2013) 55–61.

[30] A.R. Yıldız, A comparative study of population-based optimization algorithms for turning operations, Inform. Sci. 210 (2012) 81–88.