

OPTIMIZATION AND SCHEDULING OF
A POOLED LOG TRANSPORT SYSTEM

Except where reference is made to the work of others, the work described in this thesis is my own or done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

Karunakaran Haridass

Certificate of Approval:

Jeffrey S. Smith
Professor
Industrial and Systems Engineering

Jorge Valenzuela, Chair
Associate Professor
Industrial and Systems Engineering

Timothy McDonald
Associate Professor
Biosystems Engineering

George T. Flowers
Dean
Graduate School

OPTIMIZATION AND SCHEDULING OF
A POOLED LOG TRANSPORT SYSTEM

Karunakaran Haridass

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Master of Science

Auburn, Alabama
August 10, 2009

OPTIMIZATION AND SCHEDULING OF
A POOLED LOG TRANSPORT SYSTEM

Karunakaran Haridass

Permission is granted to Auburn University to make copies of this thesis at its discretion, upon request of individuals or institutions and at their expense. The author reserves all publication rights.

Signature of Author

Date of Graduation

VITA

Karunakaran Haridass, son of Haridass and Lakshmidēvi was born on May 19th, 1985, in Coimbatore, India. He entered Sri Krishna College of Engineering and Technology and graduated from the Department of Mechanical Engineering with a bachelor's degree in April 2006. In January 2007, he entered Auburn University to pursue a Master's degree in Industrial and Systems Engineering. During his years at Auburn University, he served as a graduate research assistant in the Department of Industrial and Systems Engineering. He has also worked in an aerospace project in association with the Department of Aviation and Supply chain management and the Society of Satellite Professional International Organization.

THESIS ABSTRACT
OPTIMIZATION AND SCHEDULING OF
A POOLED LOG TRANSPORT SYSTEM

Karunakaran Haridass

Master of Science, August 10, 2009
(B.E., Anna University, 2006)

105 Typed Pages

Directed by Jorge Valenzuela

The log truck scheduling problem (LTSP) under capacity constraints and time window constraints which is an NP-hard problem, involves the design of a set of best possible routes for a set of trucks starting and ending the day at a hub serving a set of loggers and mills. The objective is to minimize the unloaded miles traveled by trucks by pooling all the trucks owned by the trucking companies and optimizing the routes of the trucks. The other constraints include the speed limit, shift duration of the drivers and mandatory return of trucks to the hub at the end of the day. A deterministic simulator is developed using a C program which emulates any set of routes and gives total unloaded miles as an output besides other performance metrics. A simulated annealing algorithm is

used to generate an initial feasible solution and thereby improve the solution using various search operators functioning in the neighborhood. A sensitivity analysis is made to reduce the number of trucks needed to meet the demand for the day. For a small problem, the algorithm produces very close results to the optimum solution. A large practical problem with 68 trucks, 22 loggers and 13 mills is solved to find the best set of routes using the developed algorithm. Finally an economic analysis is made to find the impact of the location of loggers and mills on the solution. Computational results show that the number of unloaded miles is reduced by 28% and the number of trucks needed to serve the loads is reduced by 17%. The results are further verified by feeding the set of routes as input to the simulator.

ACKNOWLEDGEMENTS

I would like to express my humble gratitude to Dr. Jorge Valenzuela, my advisor and committee chair, for his support, motivation, patience and insightful guidance throughout my research work. I would like to thank my committee members Dr. Jeffrey S. Smith and Dr. Timothy McDonald for taking time out of their schedules to be on my committee. I am grateful to my parents and my sister for their love and continuous encouragement. Last but not the least, I take this opportunity to thank God for blessing me in completing the course work and research successfully.

Computer software used: Microsoft Word 2003, Microsoft Excel 2003, Google Earth,
C Language

TABLE OF CONTENTS

| | |
|--|-----|
| LIST OF TABLES..... | xi |
| LIST OF FIGURES..... | xii |
| CHAPTER 1. INTRODUCTION..... | 1 |
| CHAPTER 2. LITERATURE REVIEW..... | 7 |
| CHAPTER 3. RESEARCH BACKGROUND..... | 20 |
| CHAPTER 4. PROPOSED APPROACHES..... | 26 |
| 4.1. ASSUMPTIONS..... | 27 |
| 4.2. INTEGER PROGRAMMING APPROACH..... | 28 |
| 4.2.1. Formulation..... | 28 |
| 4.2.2. Demerits of integer programming..... | 31 |
| 4.3. SIMULATED ANNEALING APPROACH..... | 31 |
| 4.3.1. Simulated annealing implementation..... | 34 |
| 4.3.2. Construction heuristics..... | 34 |
| 4.3.3. Improvement heuristics..... | 38 |
| 4.3.4. Conditions for accepting a bad solution..... | 45 |
| 4.4. DESCRIPTION OF THE SMALL PROBLEM..... | 47 |
| 4.4.1. Solution using integer programming model..... | 47 |
| 4.4.2. Solution using simulated annealing algorithm..... | 48 |

| | |
|--|----|
| CHAPTER 5. DETERMINISTIC SIMULATOR..... | 50 |
| 5.1. BASIC MODEL OF SIMULATOR..... | 50 |
| 5.2. INITIALIZATION OF PERFORMANCE PARAMETERS..... | 53 |
| 5.3. SIMULATION OF THE EVENTS..... | 54 |
| 5.4. ADVANTAGES AND USES OF THE SIMULATOR..... | 57 |
| CHAPTER 6. EXPERIMENTAL RESULTS OF ACTUAL SYSTEM..... | 59 |
| 6.1. DESCRIPTION OF THE ACTUAL PROBLEM..... | 59 |
| 6.2. FINISHING TIME OF TRUCKS..... | 60 |
| 6.3. IMPROVEMENT IN PERFORMANCE METRICS..... | 62 |
| CHAPTER 7. SENSITIVITY ANALYSIS..... | 68 |
| 7.1. REDUCTION OF TRUCKS..... | 68 |
| 7.2. SCENARIO ANALYSIS..... | 74 |
| 7.3. MARGINAL CONTRIBUTION OF LOGGERS AND MILLS..... | 78 |
| CHAPTER 8. CONCLUSION AND DISCUSSION..... | 83 |
| REFERENCES..... | 87 |
| APPENDIX 1. LOGGER AND MILL NOTATION..... | 89 |
| APPENDIX 2. DISTANCE MATRIX BETWEEN LOGGERS AND MILLS..... | 91 |
| APPENDIX 3. DEMAND MATRIX BETWEEN LOGGERS AND MILLS..... | 92 |

LIST OF TABLES

| | | |
|-----|--|----|
| 4.1 | Solution using integer programming approach..... | 48 |
| 4.2 | Solution using simulated annealing approach..... | 48 |
| 6.1 | Improvement in performance metrics..... | 63 |
| 6.2 | Average waiting time at logger locations..... | 65 |
| 6.3 | Average waiting time at mill locations..... | 66 |
| 7.1 | Results of marginal contribution of loggers..... | 80 |
| 7.2 | Results of marginal contribution of mills..... | 81 |

LIST OF FIGURES

| | | |
|-----|--|----|
| 3.1 | Location of loggers and mills in the map..... | 21 |
| 3.2 | Tracking data given by the company..... | 24 |
| 4.1 | Pseudo code for simulated annealing..... | 36 |
| 4.2 | Decrease operator being applied to cut short the route length..... | 39 |
| 4.3 | Increase operator being applied to increase the route length..... | 40 |
| 4.4 | Stop operator being applied to stop the truck..... | 41 |
| 4.5 | Transfer operator being applied to trucks' routes..... | 42 |
| 4.6 | Interchange operator being applied to trucks' routes..... | 44 |
| 5.1 | Basic model of a simulator..... | 50 |
| 5.2 | A simulator following a truck's route..... | 52 |
| 5.3 | Schematic diagram of the simulator..... | 54 |
| 5.4 | Pseudo code for working of simulator..... | 55 |
| 6.1 | Finishing time of trucks in the existing system..... | 61 |
| 6.2 | Comparison of finishing time of trucks between real data and the best solution..... | 62 |

| | | |
|------|---|----|
| 7.1 | Finishing time of 68 trucks from the optimal solution..... | 69 |
| 7.2 | Sensitivity analysis for reduction of trucks from 67 trucks to 62 trucks..... | 69 |
| 7.3 | Sensitivity analysis for reduction of trucks from 62 trucks to 56 trucks..... | 70 |
| 7.4 | Finishing time of trucks for 68 trucks and 56 trucks..... | 71 |
| 7.5 | Time taken to find the solution with reduction in number of trucks..... | 72 |
| 7.6 | Proportion of trucks with maximum utilization..... | 73 |
| 7.7 | Six sample scenarios showing uniform distribution of loggers, mills and hub..... | 75 |
| 7.8 | Histogram of loaded miles percentage for 175 scenarios..... | 76 |
| 7.9 | Sample scenario for loaded mile 46%..... | 77 |
| 7.10 | Sample scenario for loaded mile 66%..... | 78 |

CHAPTER 1

INTRODUCTION

The log truck scheduling problem (LTSP) 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. The felled trees are first moved to a deck for processing using skidders. Processing (removal of remaining limbs and tops) is done either manually by capable log makers or by using sophisticated machines. From there, the tree-length logs are transported to the specific mills such as saw mills and paper mills.

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.

In the existing system, the logging company owns a group of trucks and operates between a particular set of mills to satisfy the demand. Some group of trucks

takes more time to meet the loads exceeding the time windows for the loggers and mills. A few other truck groups take less time to meet the demand and finish well within the time window. Some logger owners subcontract trucks to meet the demand. Very often, the trucks reach the supply point before the earliest time of operation at supply points and wait for the supplier to begin operations at the supplier site. The truck driver's working hours are not properly utilized and he spends a lot of time in just waiting for the truck to be loaded at the supplier's site.

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 point and delivery point. If trucks don't reach the pick up point at the right time, the inventory level at the logger reaches maximum 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 carry the cutting and other shaping operations. In case of non-availability of logs, the cranes and machines would be idling causing low utilization of equipment. To prevent this, the truck needs to deliver the loads at uniform intervals to ensure the smooth operation of mills.

If a group of trucks arrive together at the same destination or supplier point, they have to wait in line for a longer time to get the services of cranes for loading and unloading purposes. There is huge savings when less time is being spent in waiting and cranes are made to work all the time. This ensures continuous operation at upstream logging stations and downstream mills.

The location of loggers and mills decides the feasibility of opting for a shared transport system. If the mills and loggers are located very close to each other then no appreciable improvement can be made in sharing the trucks and delivering the loads.

The role of log transport manager is to form a demand matrix of the loads to be delivered and a distance matrix which contains the shortest path between any two locations. The mills and loggers furnish the requirements and trucks available for transport one day in advance. Then the manager works up a schedule to meet all the loads and gives it to the truck drivers to execute the schedule. Though the trucks are shared in the existing system, the routing is not optimized and leads to a large percentage of unloaded miles, more waiting time and inability to meet the demand within the time window.

In the proposed system, all the operations of the trucks are brought under a centralized ownership and the transport manager prepares a schedule taking into consideration the demand matrix of all the loggers and mills situated in that region. The centralized scheduler feeds the demand matrix and distance matrix into the computer and gets the best set of routes for all the trucks for the following day. The routes are designed in such a way that the trucks end the day before shift ends and travel fewer unloaded miles than the existing system. The income generated by the entire group of trucks is shared among the truck owners proportionately. By this way, more income is generated within a shorter period of time.

The trucks begin the day either as loaded or unloaded from the hub. They travel through a series of loggers and mills and return back to the hub at the end of the day. The

basic objective is to minimize the unloaded miles and to meet all the demands within the time window. Since there are a large number of trucks available to meet the demand, the number of trips is actually restricted so that all the trucks are completely utilized within the time window to deliver the loads. There is loading and unloading time at each logger and at each mill site.

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. But in 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 can't be easily solved using the existing heuristic approaches.

A simulator and a main program which successfully outputs the best set of routes using a simulated annealing algorithm have been developed. Basically, the simulator is a C program consisting of data structures which simulates the movement of trucks as per the set of routes given as input to the system and gives output as total unloaded miles, unmet loads and finishing time of trucks for that input.

The main program which gives the best routes as output incorporates the simulator and simulated annealing algorithm. The input given to the main program is the number of trucks, demand matrix, distance matrix and maximum number of trips allowed for the trucks. The program builds an initial feasible solution and goes on to find the best

solution by searching and improvising at various stages of the algorithm. The program is run one day in advance and actual demand to be satisfied is known one day in advance.

The logger locations and mill locations are geographically defined and the shortest distance between the loggers and mills is found using Google Earth and verified using data extracted from the data analysis of a practical case. It was found that there is only one shortest path to travel between any two customer locations.

The trucks available are enough to meet the demand requirements for the day. The trucks should not be under utilized and at the same time should not be over utilized. Therefore the number of trips traveled by the trucks is restricted to a specific value. There is a lower bound and an upper bound for the starting time and ending time for truck drivers which is fully respected. The truck is fully loaded at each logger. Hence there is no room for loading at another logger site before delivering the carrying load. The main activities performed by the trucks are to transport the loads from logger to the mill apart from starting and ending the day at the hub. The costs involved are the transportation costs between loggers and mills. The unloaded miles traveled by the trucks increases the unwanted cost.

The simulated annealing algorithm developed consists of 2 stages. In the first stage, an initial solution generated is made feasible by meeting all the demands for the day. The solution is then improved using improvement operators like transfer, exchange, interchange and stop operators. There is no particular stopping criterion employed to stop the algorithm relative to the solution change. The algorithm is allowed to run for the entire temperature steps.

The algorithm produced close results when tested against a small problem whose optimum value is known. The same algorithm without any modification is used to solve the practical case of a big problem where there are comparatively more trucks, loggers and mills. Total unloaded miles and number of trucks needed to meet the loads are taken as the measures of comparison between the existing practical system and proposed pooled transport system. It is found that the unloaded miles are reduced by 28% and the number of trucks needed to serve the loads is reduced by 17%.

The percentage of total loaded miles to the total miles traveled is a measure of the efficiency of the proposed system. In order to find the distribution and statistical mean of loaded miles percentage, the loggers and mills locations are randomly generated within a square region in a particular scenario and loaded miles percentage is calculated. This is repeated for 175 different scenarios and a distribution is obtained.

Finally the impact of individual loggers, mills and hub on the solution is found using economic analysis.

CHAPTER 2

LITERATURE REVIEW

The majority of work done previously discusses solving the vehicle-routing problem rather than a log-truck scheduling problem. Relevant literatures are being reviewed in this chapter.

Gerdesson [1] has made an attempt to solve the vehicle-routing problem with trailers. The vehicle consists of a trailer and a truck. The vehicle always starts from a depot and ends the day in the depot. The vehicle starts serving the customers one by one with both trailer and truck. After reaching a parking spot, the vehicle leaves the trailer in the parking spot and serves the nearby places with the truck alone thereby reducing the time taken to serve the customers near the parking spot. The truck returns to the parking spot, reattaches the trailer with it and serves the rest of the customers with the complete vehicle.

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.

Under construction heuristics, the number of vehicles needed to serve the loads is first calculated. Three different heuristics were used to get a good initial feasible solution. In the first heuristic, the customer locations were allocated disproportionately to the vehicles and also had the disadvantage of selecting a parking place very early. In the

second heuristic, however the locations were allocated proportionately but still the parking place was allocated very early. In the third heuristic, the parking place was not allotted and hence it provided huge potential for the improvement of the feasible solution by moving the parking place around.

At last, the improvement heuristic is used to get a good solution by exchanging customer sites between a truck's route and a vehicle's route, two truck routes, and two vehicle routes. Also some improvement is made by disconnecting the links at some location and connecting them at other customer sites.

Chao [2] models a truck and trailer routing problem [TTRP] 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. In the descent improvement step, the initial infeasible solution obtained is then converted to a feasible initial solution by exchanging one or two customers at a time and by changing the parking place without exceeding the capacity of trailer and truck.

Finally, the whole solution is improved by using the tabu search heuristic coupled with the deviation concept. In order to escape local optima and to prevent a too strict tabu restriction, intensification and diversification strategy under aspiration criteria were used to effectively search the solution space. Clean-up and stopping rules are used to improve the solution locally and the global stopping rule was used to stop the search procedure.

Sumichrast et al. [3] have compared the solutions obtained by heuristic with that of lower bound integer programming solution from LINDO, 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.

In the search heuristic, the solution was improved by exchanging plants within the truck's route and also by exchanging trucks used to meet the demand. In order to incorporate the entire network, the duplication of the depot was carried out.

The paper has proved that the heuristic method can produce the results in less than a second whereas it took more than 67 seconds to solve using LINDO. Also the gap between heuristic solution and lower bound was found to be less than 5% for a small or medium sized problem whereas it was approximately 8% for slightly bigger problems. No practical case is presented and compared in terms of improvement in objective constraints and also there were no time windows.

Thangiah et al. [4] have developed a heuristic for the vehicle routing problem with backhauls and time windows (VRPBTW). The returning vehicle after delivering the loads to the customers visits the supplier and carries the raw material along with it to the depot. In this paper, the earliest and latest time were defined which indicates 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.

The construction heuristic starts by assigning a truck to serve a seed customer initially and then including all other customers one by one without exceeding the time

windows. Then the routes for all other trucks are designed in the same fashion. When doing the above, the capacity constraints and backhaul constraints were taken into consideration.

After constructing the routes, the λ -interchange procedure is performed to improve the solution either by first improvement (or) best improvement strategy. Also, the 2-opt exchange procedure is done to further bring the solution close to optimal solution by paying attention to feasibility since inserting back haul customers within line haul customers would spoil feasibility. Finally it was found that the heuristic produced a solution which differs from the optimum only by 2.5%.

Czech et al. [5] have used parallel simulated annealing algorithm to solve vehicle routing problem with time window (VRPTW) without any backhauling. The waiting time, service time, earliest time and latest time were all taken into consideration while solving the problem. The objective function had to minimize the number of vehicles used as well as to reduce 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.

The paper also discusses the criterion to select the initial temperature and also the termination criteria of when the search should be stopped. Excellent conclusions on what should be done to get good optimization results were summarized. Proper care should be taken when selecting initial temperature for annealing, cooling schedule, number of annealing steps executed at each temperature, along with relative importance of number of routes against the travel distance.

Ronnquist [6] 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, L.P. methods, branch and bound methods, heuristics and column generation were used for optimization.

In the transportation and routing stage, transportation planning was done in one of three ways. Either the individual truck owners decide the routing or routing is carried out by the harvesting people once the wood accumulates. The third case is the integration of all the truck owners under a centralized system, and combined routing of all the trucks is carried out by a single virtual owner.

The paper clearly points out the difference between general vehicle routing problems and routing problems in forestry. The difference is that in case of routing in forestry, there are many supply and demand points. The supply and demand locations are to be visited more than once unlike the case of general vehicle routing problems. Usually the solution time is 5 seconds for truck dispatching and 1 day for truck scheduling. The solution approach here consisted of both heuristics and optimization.

Bent et al. [7] have modeled the dynamic vehicle routing with stochastic service times and customer locations using Multiple Plan Approach (MPA) and Multiple Scenario Approach (MSA). Multiple plan approach's main idea is to generate and maintain all possible sets of routing plans along with the main routing plan. The random customer requests are ranked according to a priority. The requests are checked against the

available plans. If they match one of the available plans, it is accepted; otherwise it is rejected.

In case of multiple scenario approach, the requests are sampled from a general log normal distribution and the routing plans are generated according to sampled requests. Hence the future customer requests are accommodated by switching from the main routing plan to the available plan. As an improvement to this, the multiple scenario approach with loosely constrained problems (MSA-LC) were modeled. In this type, the trucks delay the departure from the customer location anticipating any request from the nearby customers. This approach yielded good results. Finally, the consensus model is evolved which generates routes according to MPA and MSA-LC and executes the plan which yields the best results.

Weintraub et al. [8] 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. To achieve this, the supply point for the first trip and the demand point for the last trip were selected near to the driver's home. The demand for one of the product types can be satisfied by any of the supply points and this relaxation has imparted a large amount of flexibility in meeting the demand. The scheduling is designed in such a way that the inter-arrival time is almost uniform for all the trucks so that the production and the harvesting operations are done smoothly at either end without excess inventory or a lack of inventory.

Scheduling is done every 1 hour taking into consideration the waiting time of the trucks, idle times of the cranes at the source and at the destination, and also the trucks

that will be idle during and after 1 hour. This method will help in reducing competition among the truck drivers in competing for loads and also ensures equal income to all the truck owners. 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. [9] proposed a solution approach based on column generation and pseudo branch and price for the log-truck scheduling problem. A cluster of customer locations near the starting point are only considered for destination in that trip. Also the truck is allowed to visit all the supply points until it is fully loaded before beginning to deliver the loads to the demand locations. The truck operates in two 8-hour shifts with time allocated for lunch break. The sub-problem of designing a route for the truck was solved using the shortest path method.

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. While solving the sub-problem using k-shortest path method, a series of shortest path routes are generated so that if the first one is not feasible, the next shortest path is automatically adopted. Also the solution for the sub problem sets a lower bound for the problems after interrupting the optimality with a fixed number of iterations. Finally the column generator is combined with fast heuristics to produce efficient routes for the log-truck scheduling problem.

Hirsch et al. [10] have used four different tabu search approaches and a few post optimization heuristics to solve the timber transport vehicle routing problem (TTVRP).

The first three approaches were intended for small problems while the fourth one is meant for big problems. In the case of small problems, the approaches differ in terms of search in the neighborhood. One standard tabu search (TS) approach searches a large neighborhood whereas the tabu search with limited neighborhood (TSLN) approach searches a very limited neighborhood. The third approach is a combination of the above two approaches with full neighborhood search done after a fixed number of limited neighborhood searches.

Though every approach's objective is to minimize the empty truck movements, the tabu search with large problem instance (TSLP) attempts to solve larger problems by limiting the task allocation to 4. By doing this, the constraint of time window is satisfied at all customer and destination locations. Even though all approaches produce good results, it was found that the solution got stuck to local optima most of the time. Hence a post-optimization improvisation heuristic, namely 2-opt, was done to produce a much better solution. It was observed that the heuristics produced a result closer to the lower bound produced by other CPLEX solvers.

Bent et al. [11] discuss the importance of solving vehicle routing problems with time windows (VRPTW) 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 is succeeding only in minimizing the travel cost.

While minimizing the number of routes, traditional move operators like 2-exchange, or-exchange, relocation, crossover, and exchange were used. The operators and customers to be operated upon were chosen randomly. The objective function is

modified so that the solutions that have routes with more customers and routes with fewer customers are more favored over solutions that have routes with equal distribution of customers. The limited discrepancy search (LDS), a strategy which explores the search tree in waves allowing the heuristic to make more mistakes is used in minimizing the travel cost. This strategy yielded a good optimal solution after appreciable iterations. The results were then compared with the standard Solomon benchmarks. The 2-stage algorithm produced matching or better results 97% of the time, the best percentage of improvement being 2%.

Osman [12] has investigated three different algorithms namely descent, hybrid simulated annealing with tabu search and tabu search algorithms for their performance in solving vehicle routing problem (VRP) with capacity and the distance constraints. The well known Clarke and Weight savings procedure was used to find an initial feasible solution. The λ -Interchange generation mechanism was used to search the neighborhood of the current solution. The mechanism consists of the shift operator and the interchange operator. The first improvement (FI) and the best improvement (BI) strategy were used to select alternative solutions. The λ -interchange descent algorithm has a major limitation of dependence of the final solution on the initial feasible solution. Hence the final solution obtained may be far away from the global optimum. The best available selection (BA) strategy and the first best available selection strategy (FBA) have been used to select alternative solutions. A special data structure was created for the best available selection strategy. This has helped in reducing the computational time by 50%. The simulated annealing algorithm produced results better than descent algorithm but the variance of the

quality of the solution and that of the computational time was found to vary significantly. Tabu Search was the best of all algorithms used but it requires a lot of storage space for selection of the alternative solution.

Valenzuela et al. [13] have designed a transportation scheduling system for silvicultural projects considering project due dates, precedence relationship and transportation costs. There were five different classes of resources (trucks) and each had their own routes. The solution approach consists of 2 stages starting with simulating the network and then improving the solution using search heuristics. The simulation model outputs the completion time for tasks and also the distance traveled by the resources. Under simulated annealing, the search operators engaged in the search technique are the switch operator and the transfer operator. The objective of minimizing the project duration and minimizing the travel distance was considered separately and dually as well. Sensitivity analysis was done by varying the penalty cost for minimizing project durations and the penalty cost for transportation. Under resource capacity planning, marginal analysis was being done to find the increase or decrease in cost by adding or removing one resource from each of the resource class one at a time.

McDonald et al. [14] have compared the efficiency of a shared transport system with that of the existing traditional log transport system. The paper dealt with issues of potential savings, scale of operations and dispersion pattern of loggers and mills. The total unloaded miles were used as a measure of efficiency between logger-controlled transport system and pooled transport system. In the case of a logger-controlled transport

system, the unloaded miles constitute just half the total distance traveled by the trucks. The speed constraints and the shift duration constraints were brought into the model.

A test case with 16 loggers and 2 mills randomly distributed over a square region of 250 miles was used. It was found that a minimum number of loggers was needed to achieve maximum efficiency above which the savings remained unchanged. Also when the mills were located a bit far away, more loggers were needed to achieve maximum efficiency.

A linear increase in the transportation efficiency was observed when the range of logging sites increased. Also an increase in the range of distribution of loggers increased the average of efficiency gained but the variance was very high which may possibly reduce the benefits gained.

Simulated annealing is a meta-heuristic algorithm used for optimizing combinatorial problems. It is mainly used to solve truck routing, school bus routing, scheduling etc. The name comes from the annealing process in the heat treatment of solids to reduce defects and to reach a configuration with reduced internal energy. Before heating the solid, the atom possesses high internal energy in the solid. When it is heated, the atoms are free to move and wander inside the solid. The solid is allowed to cool slowly and the atoms occupy a desired state with lower internal energy.

Similarly, in the case of combinatorial problems, the objective is to reach the global optimum which is assumed to be a state of lower internal energy than the existing state. For routing problems, the trucks are allowed to move freely during the initial stages of high temperature. They are then allowed to settle down with the best possible routes by

searching through the neighborhood during the later stages of the simulated annealing algorithm.

The algorithm searches for a good solution throughout the feasible region during the initial stages without getting stuck to the local optimum. This ensures that the part of the region which contains the global optimum value is not missed. But as the temperature goes down, the ability to search and find a solution close to the global optimum also goes down since it gets stuck in some neighborhood. The rate of cooling is critical in helping the algorithm to achieve the optimum solution.

An evaluation function is used to find a value corresponding to the new solution generated. The function decides how good or bad the alternative solution is. At high temperatures, even a small increase in the value of the evaluation function is accepted with a probability that depends on the value of the temperature and the difference in the value of the evaluation function. The acceptability of a bad solution goes down with reduction in temperature. It means that the bad solution is rarely accepted at lower temperatures. The majority of search towards the end is oriented towards finding a good solution around the global optimum.

The simulated annealing algorithm is very flexible and can accommodate any non-linear model and difficult constraints applicable to any problem. The algorithm is very robust and capable of approaching the global optimum value. Depending on the nature of the problem, proper selection and execution of the search heuristics are possible. The algorithm is designed to look for solutions to the problems rather than modeling it.

The closeness of the results obtained from simulated annealing to the global optimum depends on the computational time. The more time it takes to compute a solution, the closer is the solution to the optimum value. Care should be taken to properly define the constraints in the algorithm. The penalties involved in the evaluation function and the parameters for the annealing process should be defined after a trial and error method. The algorithm does not take with it the knowledge obtained after searching. Instead, it searches for a good solution in the entire feasible region. The algorithm cannot be used whenever there is an equality constraint. Also the algorithm is not useful when our interest is to find a best solution in more than one feasible region separated by an infeasible region.

Integer programming can be used to solve the routing problem but the time taken to solve the problem increases exponentially with the size of the problem. Nevertheless, integer programming can still be used to find a good lower bound for the optimum solution. The lower bound can be compared with the solution obtained using the simulated annealing algorithm to find out the efficiency of the algorithm. Still integer programming needs strong formulation and any error in it may lead to an infeasible solution. More time is spent on modeling the problem rather than solving it. Even with advances in the processing speed, the time to solve the problem has not come down appreciably. It still takes more than one or two days to solve the problem and to obtain the best set of routes. Industry people expect good solutions not necessarily the optimum within 10 min so that the decisions can be made accordingly.

CHAPTER 3

RESEARCH BACKGROUND

This research was motivated by an existing log truck transport system covering a couple of southern states in the US. There were high costs involved in transportation from logger to mill. The current system was less efficient with more trucks being used to transport logs. The drivers work for more than 12 hours to meet the demand leading to overtime and causing more expense for the management. The trucks were made to wait for a long time in queue and, thus, were not properly utilized in spending more time in transporting logs to the mill. Due to improper planning of the routes, the trucks travel more unloaded miles which must be reduced to increase the profit of the operation.

The existing system has 22 loggers, 13 mills and 2 hubs spread in and around Louisiana and Mississippi. Figure 3.1 shows the distribution of loggers and mills on the map. The balloon shaped lettered ones are the mills and the flag shaped numbered ones are the loggers. The loads are required to be transported from the loggers and delivered to the mills. The trucking company own about 68 trucks that carry the loads and meet the demand in any particular day. The main problem with this system is that trucks always serve a particular closed set of loggers and mills. For example, a truck will serve the loads only between logger 1 and mills 1, 2 & 3 and never really serves the loads between remaining loggers and mills.

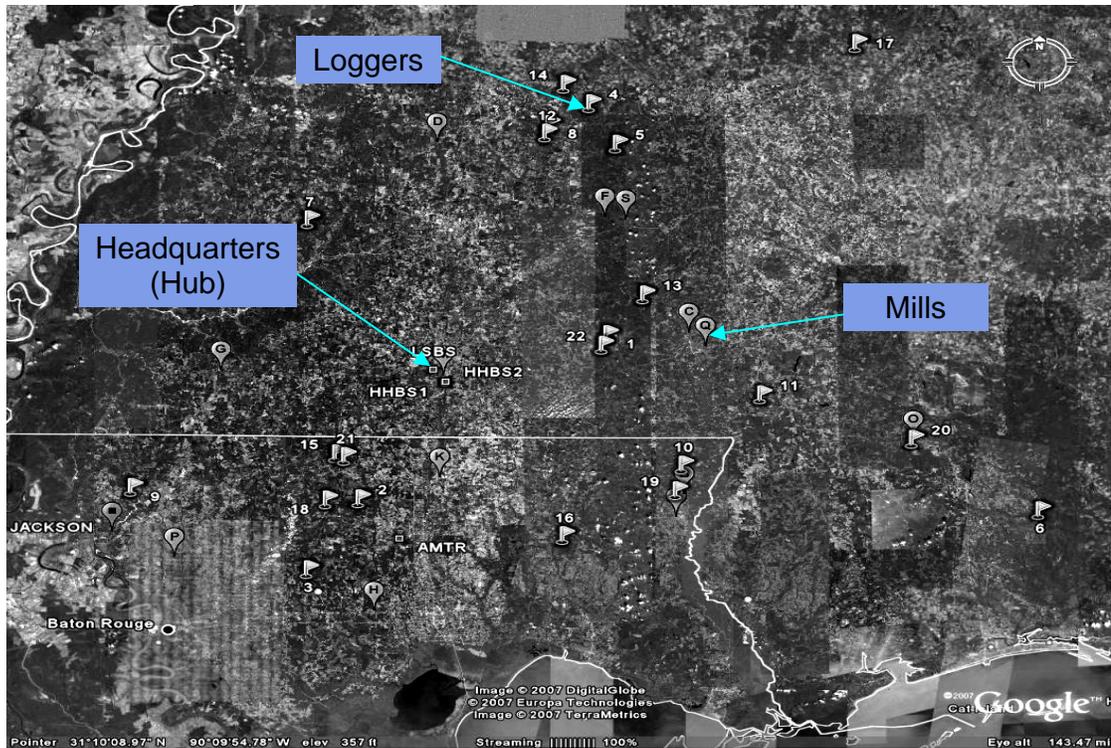


Figure 3.1: Location of loggers and mills in the map.

As a result, some trucks which have fairly fewer loads to meet remain idle most of the time, whereas other trucks which should transport more loads take extra time and travel much longer distances in order to meet the loads and with more unloaded miles and improper resource utilization.

The route optimization is done for all the trucks under pooled transport system to bring an optimal solution thereby reducing the unloaded miles and avoiding excess time by properly utilizing the trucking resources. The trucks always start from the hub and end the day at the hub. In the existing system, though the average ending time is 7 hours, the highest ending time of all the trucks is 18.5 hours, indicating high variance. Maximum number of trips allowed is 7 which is very high. The truck drivers are paid for overtime

hours exceeding the normal shift duration. The trucks were found to be spending more time waiting at logger and mill sites.

Our main objective is to reduce the cost associated with the existing system. The costs are incurred in the form of more unloaded miles, excess time taken to meet the demand and in usage of more trucks than needed.

A truck should always start from the hub and end the day at the hub. The truck can be either loaded or unloaded depending on the ending status of the truck on the previous day. Due to the shift duration of 8 hours, the truck may end the previous day as loaded. Hence it should start the following day as loaded and move towards the mill as the first destination to deliver the load. Then the truck visits the logger to load the truck and the cycle continues thereafter. Most of the trucks start the day as unloaded. The unloaded trucks would have delivered the load at the mill before returning to the hub on the previous day and would proceed first to a logger to be loaded. Then the truck moves on to the mill and the cycle continues.

The maximum number of trips is restricted to five to account for the time window of the driver's shift hours, and the starting time and the closing time at the loggers and mills. This would allow for all the trucks to be utilized efficiently for the full time of operation.

The shift duration is restricted to 8 hours so that the drivers return to the hub in time and the cranes at the logger and mill sites can finish the operations well within the time window. The truck is always fully loaded at the logger. Hence there is no chance to load for the second time before unloading the current load. The speed of the truck is

limited to 50 miles/hour. The truck can choose any mill as its destination starting from the logger provided there is an unmet load requirement between the selected pair of the mill and logger.

In a single trip, a truck can visit one logger and a mill at a time. After delivering the loads in the mill, it can then choose to load at any logger provided there are loads to be carried from that particular logger to any mill. The total number of loads delivered between the logger and mills should be exactly equal to the load requirement between the loggers and mills at the end of the day. In short, there should be no unmet loads or over served loads.

It is assumed that all the trucks have to return to the hub at the end of the day. There are 2 hubs located very close to each other. In this problem 2 hubs are combined into one since the distance separating them is less than 2 miles.

The trucks are loaded and unloaded at the customer locations based on first in first out servicing. The truck has to follow a queue if there is one at the logger and mill before getting access to the logger or mill. The waiting time of trucks in queue is unwanted and must be minimized or eliminated completely so that the truck's time can be better utilized in transporting logs from the loggers to the mills.

The mills may need several different types of wood from the loggers. Trucks are required to deliver those specific types of wood to the mills. This is actually taken into consideration when drawing up a demand matrix which addresses the number of loads of different types of wood that are to be transported to the mill. The size of the truck is sufficient to transport all types of wood between a logger and a mill.

The trucks are assumed to be homogeneous and can have access to any logger and mill irrespective of the location of loggers and mills in the region. This assumption breaks the restriction imposed on the traditional management system which is more inclined to operate the trucks between a particular set of loggers and mills.

| DRIVER | DATE | DEPART TIME | ARRIVE TIME | TRANSIT TIME ON SITE | TOTAL TRIP TIME | IDLE TIME | MILES DRIVEN | EXCEPT | LOCATION | PRIMARY CATEGORY |
|------------|-----------|-------------|-------------|----------------------|-----------------|-----------|--------------|--------|---|------------------|
| 3 O'Neal 1 | 22-Mar-07 | 4:39 | 4:45 | 0:06 | 0:12 | 0:18 | 0:12 | 0.2 | Harvest Haul Base 1064 Lacy St Magnolia, MS 39652 | HQ |
| | | 4:57 | 6:37 | 1:41 | 0:12 | 1:53 | 0:12 | 84 | J & K 3/27/07 1153 S Ms-13 [MS-13] Brooklyn, MS 39425 | Prohibited |
| | | 6:49 | 6:51 | 0:02 | 0:14 | 0:16 | 0:14 | | J & K 3/27/07 1153 S Ms-13 [MS-13] Brooklyn, MS 39425 | Prohibited |
| | | 7:05 | 7:07 | 0:02 | 0:10 | 0:12 | 0:10 | | J & K 3/27/07 1153 S Ms-13 [MS-13] Brooklyn, MS 39425 | Prohibited |
| | | 7:17 | 9:18 | 2:01 | 0:02 | 2:03 | 0:02 | 91 | Amite Truck Stop 30214 LA-16 Amite, LA 70422 | HQ |
| | | 9:20 | 9:22 | 0:02 | 0:02 | 0:04 | 0:02 | | Amite Truck Stop 30214 LA-16 Amite, LA 70422 | HQ |
| | | 9:24 | 9:32 | 0:08 | 0:02 | 0:10 | 0:02 | 2 | 32504 LA-16 Amite, LA 70422 | Unknown |
| | | 9:34 | 10:12 | 0:38 | 0:02 | 0:40 | 0:02 | 18 | Holden Mill 30377 Edna Kinchen Rd Holden, LA 70744 | Customers |
| | | 10:14 | 12:29 | 2:15 | 0:14 | 2:29 | 0:14 | 118 | J & K 3/27/07 1153 S Ms-13 [MS-13] Brooklyn, MS 39425 | Prohibited |
| | | 12:43 | 12:47 | 0:04 | 0:06 | 0:10 | 0:06 | | J & K 3/27/07 1153 S Ms-13 [MS-13] Brooklyn, MS 39425 | Prohibited |
| | | 12:53 | 13:13 | 0:20 | 0:06 | 0:26 | 0:06 | 11 | 295 MS-13 Lumberton, MS 39455 | Unknown |
| | | 13:19 | 14:02 | 0:42 | 0:02 | 0:44 | 0:02 | 31 | 63137 LA-10 Bogalusa, LA 70427 | Unknown |
| | | 14:04 | 14:08 | 0:04 | 0:02 | 0:06 | 0:02 | | 63077 LA-10 Bogalusa, LA 70427 | Unknown |
| | | 14:10 | 15:48 | 1:39 | 0:02 | 1:41 | 0:02 | 69 | Holden Mill 30377 Edna Kinchen Rd Holden, LA 70744 | Customers |
| | | 15:50 | 17:07 | 1:16 | 0:06 | 1:22 | 0:06 | 55 | LS Trucking Base / fuel stop Aimort Rd | HQ |

Figure 3.2: Tracking data given by the company.

The trucking company provided the data which contains the movement of the trucks from the loggers to the mills in a single day for all the 68 trucks as shown in figure 3.2. The truck's movement is tracked by the satellite through the GPS equipment provided with all the trucks. Whenever a truck stops in a particular place for more than 2

minutes, its position is recorded and posted to the Excel file. The data contains the departure time, arrival time, travel time, waiting time and loading/unloading time at the site. The statistics associated with this data will be very useful when we develop a simulator and test it with a practical case to check for the output from the simulator. The demand matrix and the distance matrix for the research were extracted from the given data. The addresses of the loggers, mills and the hub were extracted and then located precisely in a map using Google Earth software. After scanning through the tracking information, it was found that the trucks were repeatedly operated between a particular set of the loggers and the mills. Hence our main aim is to reduce the total unloaded miles traveled by the truck by making the truck to serve all possible loggers and mills.

CHAPTER 4

PROPOSED APPROACHES

In a pooled transport system, all the loggers and the mills are brought under the centralized leadership of a single owner. The owner plans and schedules the movement of the trucks to meet the demand on any given day. The network consists of a hub, a series of loggers and the mills distributed near the hub. The loggers are the locations where the cut down trees are temporarily stored, and serve as the supply points of wood. The mills require loads of wood for further processing and they form the demand points.

a) CUSTOMERS:

The loggers and mills are the customers for the trucks. They are located far away from each other. Good planning is required to meet the transportation requirements between them.

b) VEHICLE:

The vehicles used are the trucks. They visit the loggers and the mills by traveling the shortest possible distances, and delivering the loads in the quickest time. The truck is driven by a driver who takes the truck at the beginning of the shift, and leaves the truck in the hub at the end of the shift.

c) ROUTING PLAN:

The routing plan consists of a set of routes assigned to all the trucks. Each truck has to follow the route given to it in picking up and delivering the loads. The routes should be designed in such a way that the truck travels a minimum of unloaded miles and at the same time should meet all the loads at the end of the day.

d) TIME WINDOW:

The normal duration of a shift is 8 hours. The truck driver and the operators at the logger and mill locations are required to work within the time window. Hence the loads have to be delivered within this time window to all the loggers and the mills. This is specifically done to overcome the labor costs associated with the overtime duty.

4.1 ASSUMPTIONS:

The locations of the loggers and mills are assumed to be vertices in a Euclidean space. They can be represented by the x-distance and the y-distance in a two dimensional space. 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. All the trucks have to start from the hub and should return to the hub at the end of the day. There is no parking space near the customer's site. For the ease of the routing and management of the trucks, the trucks will end the day in 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 requirement for that customer. The customer can have one or more demands for the load or may require more loads to be delivered to other

customers. All the trucks are assumed to have an identical capacity. The trucks always travel between the customer locations as fully loaded or as empty.

Two approaches were actually followed to solve the problem. They are

1. Integer programming approach
2. Simulated annealing approach

4.2 INTEGER PROGRAMMING APPROACH:

The integer programming approach is used to find an optimal solution to the problem. It is very useful to solve small size problems.

4.2.1 FORMULATION:

It is necessary to understand the terms used and their representation in order to understand the formulation of the objective function and their constraints.

NOMENCLATURE

T - Number of trucks

M - Number of mills

L - Number of loggers

N - Number of trips

I - Denotes truck i ($i=1, \dots, T$)

m - Denotes mill m ($m=1, \dots, M$)

l - Denotes logger l ($l=1, \dots, L$)

j - Denotes trip j ($j=1, \dots, N$)

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'.

1) Objective formulation:

The main aim of the objective function is to reduce the total unloaded miles traveled by the truck. It can be formulated as

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

2) Constraint formulation:

There are basically six constraints which define the problem space and should not be violated in solving the problem.

a) On any single trip, the truck can travel to the mill only once.

$$\sum_{l=1}^T \sum_{m=1}^M L_{ilmj} \leq 1 \text{ for all } i=1\dots T \text{ and } j=1\dots N$$

b) On any single trip, the truck can travel to the logger only once.

$$\sum_{l=1}^L \sum_{m=1}^M U_{ilmj} \leq 1 \text{ for all } i=1\dots T \text{ and } j=1\dots N$$

c) The truck can travel from a logger to any mill independent of the mill where it is coming from.

$$\sum_{k=1}^L U_{imkj} \leq \sum_{k=1}^L L_{ikmj} \text{ for all } i=1\dots T, l=1\dots L, m=1\dots M \text{ and } j=1\dots N$$

- d) The truck can travel from a mill to any logger independent of the logger where it is coming from.

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

- e) 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.

$$\sum_{i=1}^T \sum_{j=1}^N L_{ilmj} = S_{lm} \text{ for all } l=1 \dots L, m=1 \dots M.$$

- f) The average speed of the trucks is calculated as 50 mph. Therefore, since there is a shift duration of 8 hours, the truck can cover a maximum distance of 360 miles with the average speed.

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

- g) L_{ilmj} and U_{ilmj} are the binary variables (0,1)

The formulation of the objective function and the constraints is with reference to McDonald *et al.* [14].

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

On a particular day, the loads that are to be transported between the loggers and mills are fixed. The routes are to be designed in such a way that all the required loads are transported between the loggers and the mills.

The trucks should be utilized to the maximum extent possible. The routing must be designed in such a way as to reduce the number of trucks used. Fewer trucks with high utilization is better than more trucks with low utilization. Also the total distance traveled by all the trucks should be minimized. This happens only when we can reduce the total unloaded miles since 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.

4.2.2 DEMERITS OF INTEGER PROGRAMMING:

Integer programming is mainly used to solve small problems within a reasonable time. In the case of large problems, the time taken to solve the problem increases exponentially with the size of the problem. Consequently, it takes days to solve the truck routing problem which is prohibitive for the industrial people. The people working in the industry need solutions not necessarily an optimum one but are happy to have a solution very close to the optimum value within a short time. The solutions obtained using the meta-heuristics like the simulated annealing algorithm are very close to the optimum value and can be found within a short time period of 5 minutes. Hence the truck routing problem is solved using the simulated annealing approach.

4.3 SIMULATED ANNEALING APPROACH:

Simulated annealing (S.A) is a meta-heuristic algorithm which is used to solve complex large problems having a large solution space producing results close to the

global optimum value in a short period of time. The heuristic is inspired by nature and is derived from statistical mechanics. The simulated annealing algorithm was first invented 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 the n-body systems. In the later part of the 20th century, the concept of simulated annealing was put forward independently by S. Kirkpatrick, C. D. Gellat and M.P Vecchi in 1983 and also by V.Cerny in 1985. A detailed explanation can be found in Reeves (1995).

The simulated annealing algorithm is generally used to solve combinatorial optimization problems. The term annealing is associated with the heat treatment of solids. The solids are initially heated to a high temperature and then it is allowed to cool down slowly in order to impart hardness to the solid. As the solid is heated, the atoms are dislodged from their positions and allowed to roam freely inside the solid. During the cooling process, the solid is allowed to stay at that state for a sufficient amount of time at all temperatures. Hence this allows the atoms to move around freely and occupy a comfortable position. When the atoms are free to occupy any position inside the solid, then that solid achieves its maximum hardness. Finally when the solid is completely cooled, the atoms stick to that position inside the solid which gives maximum hardness to the solid.

The analogy between the annealing process used in heat treatment, and the simulated annealing (S.A) algorithm used to solve combinatorial optimization problem is best explained below. In S.A, the process starts at high temperature with an initial solution. The temperature is decremented step by step, and at each temperature search is

done for a particular number of iterations. Here the number of iterations is similar to the amount of time the solid is allowed to stay at that temperature. The best solution found so far is close to the optimal solution and is compared to the best configuration that the atoms can take in the solid which gives maximum hardness to the solid. The solid state at each temperature is similar to the feasible solution that is obtained at each temperature step.

The objective is to reach the global optimum value in the case of the simulated annealing algorithm whereas it is to reach a solid state with maximum energy in the case of the heat treatment. Rapid quenching is synonymous with the local optimization.

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

STEP 1: An initial solution is generated using the construction heuristics. The algorithm starts by fixing the temperature T =Initial temperature. To start with, the initial solution is accepted as the best solution.

STEP2: The solution's neighborhood is searched using the different search operators in the improvement heuristics.

STEP 3: The new solution is fed as an input to the simulator to get the performance measures and the value of the fitness function is calculated using these measures.

STEP 4: Depending on the value of the fitness function, the new solution is adopted as a better solution. Even if the newly obtained solution is bad, it is still accepted with a probability which depends on the current temperature and the difference

in fitness function between the bad solution and the previously accepted solution.

STEP 5: The best solution obtained so far is kept in memory and is compared with the new solution. If the new solution is better than the best solution obtained so far, then the new solution is adopted as the best solution.

STEP 6: The value of the temperature is decremented by a small step. The value of the temperature is checked against final temperature for it has reached its final value. If not, then go to STEP 2 and the steps from 2 to 6 are repeated.

STEP 7: When the final temperature is reached, the best possible solution is obtained which is very close to the optimal solution.

4.3.1 SIMULATED ANNEALING IMPLEMENTATION:

The simulated annealing algorithm consists of two main heuristics. An initial solution is first generated using the construction heuristics and then improved further using the improvement heuristics to obtain a best possible solution.

The parameters for the simulated annealing process for our system are as follows

- a) Initial temperature = 100
- b) Number of searches = 2000
- c) Decrementing factor (alpha) = 0.9
- d) Final Temperature (beta) = 0.1

4.3.2 CONSTRUCTION HEURISTICS:

The construction heuristics involve random filling of the routes of the trucks with the loggers and the mills. The number of loggers, mills, trucks and the

maximum number of trips are well known. There are two kinds of trucks: the loaded and the unloaded trucks. The loaded trucks always start the day by going towards the mill and unloaded trucks start the day by visiting a logger. Hence the first trip should always be a mill for a loaded truck and it should be a logger for an unloaded truck. Once a truck visits a logger, it would then visit the mill and then logger and so on. The loggers and the mills repeat each other alternately one after the other in the route of the truck.

Figure 4.1 contains the pseudo code for the simulated annealing meta heuristic algorithm. There are two types of matrices, namely the route logger matrix and the route mill matrix with the number of rows equal to the number of trucks and the number of columns equal to the maximum number of the trips. The route logger matrix and the route mill matrix collectively form the routes of all the trucks, the initial solution to start the algorithm. The route logger matrix is filled by the uniform random sampling of the loggers, and similarly the route mill matrix is filled by the uniform random sampling of the mills for each truck. The last trip of the trucks is filled with the hub since all the trucks should return to the hub at the end of the day.

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

Figure 4.1: Pseudo code for simulated annealing.

For the loaded trucks, there is a restriction that the truck should move to a particular mill depending on the truck ID and the destination of the carrying load.

Whereas for the unloaded truck, the truck is free to move to any logger to start the day. The initial solution need not always be a feasible solution. The initial solution is then fed to the simulator to obtain the performance metrics like the unmet loads, over met loads, unloaded miles traveled by the trucks, waiting time of all the trucks and the trucks that have not completed the trips within the time window. The fitness function is then calculated by allocating a penalty to each of the performance measures.

Fitness function (F) = Travel Penalty*unloaded miles + Penalty* unmet loads +Waiting
 Penalty *total waiting time + Excess Penalty*Over met loads +
 Hub Penalty * (Trucks that have not completed the trips)

$$\text{Fitness (F)} = K1*(UM) + K2*(UL) + K3*(OL) + K4*(WT) + K5*(NT)$$

Where,

UM - Unloaded miles

UL - Un-served loads

OL - Over served loads

WT - Waiting time

NT - Number of trucks not returning to the hub

Penalty Values:

K1 = \$1/unloaded mile

K2 = \$10,000/unmet load

K3 = \$10,000/overserved load

K4 = \$ 0.6/hr of the waiting time

K5 = \$1000/ truck not returning to the hub before 8 hours

The value of the fitness function always gives a measure of the quality of the solution: The lower the fitness, the better the solution. Hence if a solution has a lower fitness value than other solutions, then it is considered a better solution.

4.3.3 IMPROVEMENT HEURISTICS:

The improvement heuristics play a major role in finding a good solution by starting with a poor initial solution. It consists of using many operators to disturb the old solution and creating a new one. The operators are used collectively or individually based on the performance metrics of the old solution. Even after disturbing the old solution, care must be taken to ensure that the last trip is always the hub. The following are the operators that are used to improve the initial solution. They are

- a. Decrease route
- b. Increase route
- c. Stop truck
- d. Transfer route
- e. Interchange route

a) DECREASE ROUTE:

The decrease route operator is used whenever there are over met loads. Over met loads are unwanted loads that do not come under the demand matrix and the trucks are spending more time in delivering them which is unnecessary. Moreover the over met

loads make the solution infeasible. Therefore a truck and a trip are randomly selected and the loggers and mills in that selected trip are deleted.

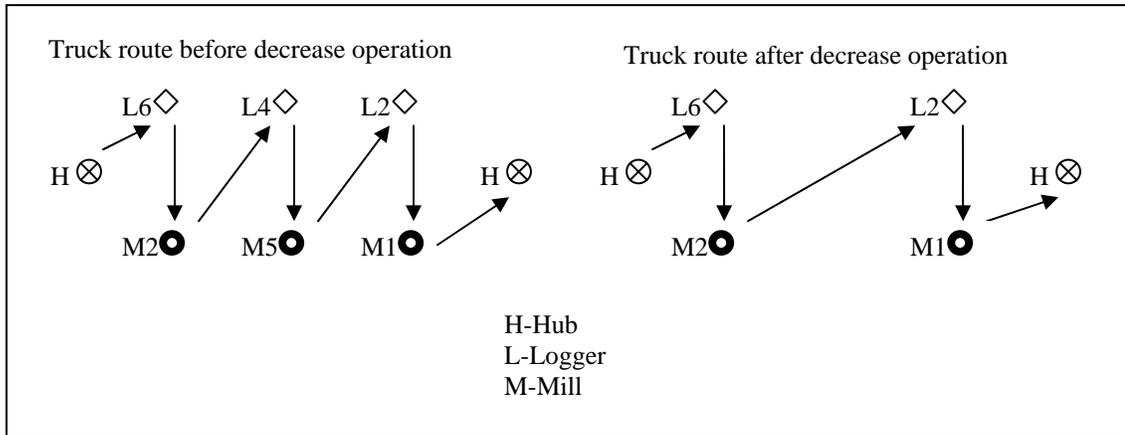


Figure 4.2: Decrease operator being applied to cut short the route length.

Figure 4.2 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 trips. 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.

b) INCREASE ROUTE:

The increase route operator assigns more loads to be delivered by the truck. This operator is used whenever there are more unmet loads. The unmet loads are those loads in the demand matrix which are not delivered during the whole day’s operation. On any cost, the loads in the demand matrix should be met and if there are any loads that are not met; the length of the routes should be increased to meet those loads. Hence a truck and

its trip are selected randomly and a pair of the logger and a mill is inserted between the previous trip and the current trip.

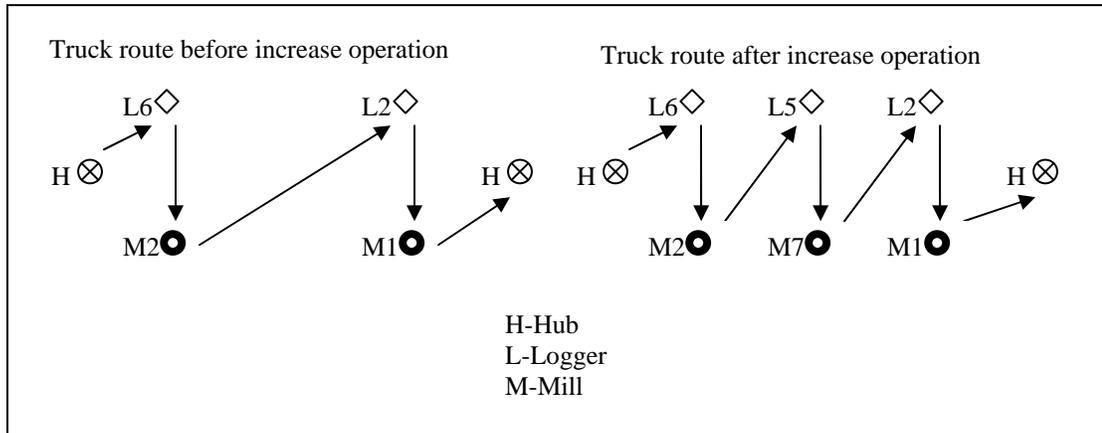


Figure 4.3: Increase operator being applied to increase the route length.

Figure 4.3 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 between the two trips. To use this operator, the truck should be running for fewer of trips than the maximum trips so that insertion in the truck’s route is possible. Again the process of increasing the route is different for a loaded truck and an unloaded truck, and the sequence of the route is respected. Increasing route operation is the opposite of decreasing route operation and is executed depending on the unmet loads or the over served loads in the solution.

c) STOP TRUCK:

Even though all the loads are perfectly met, the time it takes to meet the loads is very high. It is necessary to stop before 8 hours in order to respect the time window of an 8-hour shift for the driver and as well as at the logger and the mill locations. This operator is used whenever there is one or more trucks that have not returned to the hub

within 8 hours. The stop truck operator stops the truck by randomly selecting a trip and deleting the loggers and the mills in that trip and in the following trips.

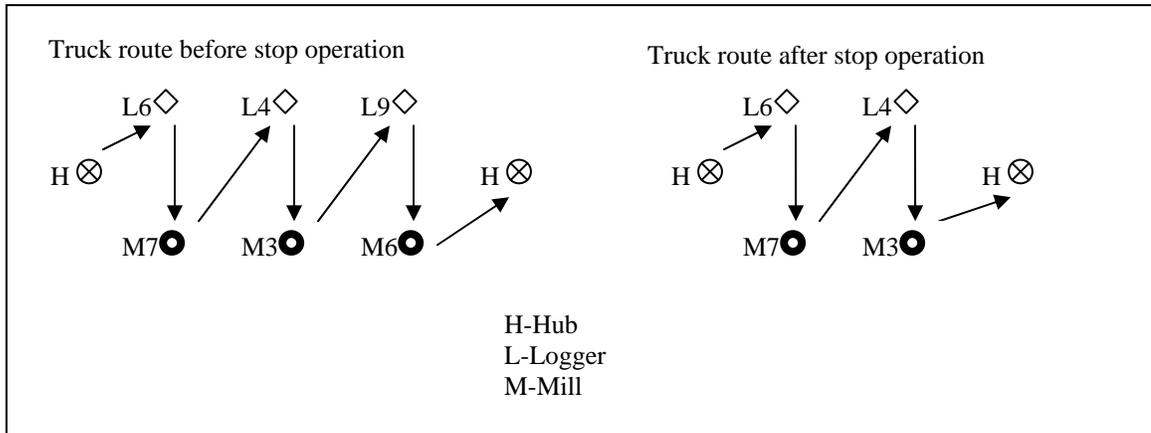


Figure 4.4: Stop operator being applied to stop the truck.

Figure 4.4 shows the truck's route before applying the stop operation. The last trip which consists of logger 9 and mill 6 is removed from the truck's route during the operation. 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, you 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 is again made feasible using the Increase operator or the Decrease operator.

d) TRANSFER ROUTE:

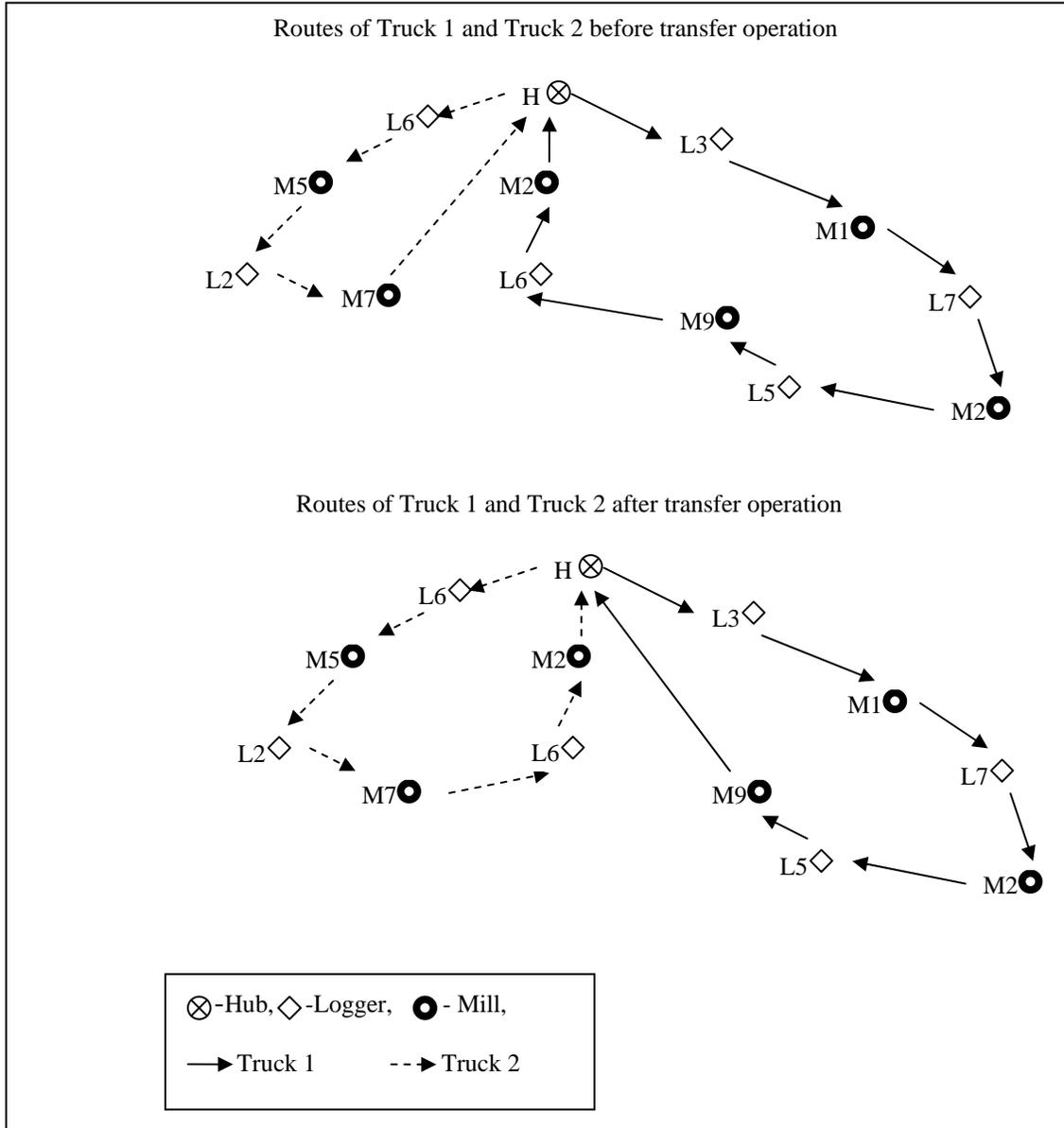


Figure 4.5: Transfer operator being applied to trucks' routes.

Figure 4.5 shows the routes of the trucks before and after applying the transfer operator. The transfer route operator is used to transfer the trip from one truck's route to another truck's route or to another part of the same truck's route. This operator reduces the total unloaded miles traveled by the truck. Two trucks and two trips are randomly

selected. The logger and the mill from the first truck's route is transferred to other truck's route and inserted after the randomly selected route. The last trip of logger 6 and mill 2 is transferred from the truck 1's route to the last trip in truck 2's route during the transfer operation. Earlier truck 1 had 4 trips 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 delivering the loads. The feasibility of the solution is not affected since this is just a transfer operation from one truck's route to another truck's route. The unmet and over met loads will remain unchanged whereas the truck may sometimes take a longer time to meet the loads.

e) INTERCHANGE OPERATOR:

The trucks perform a pair wise exchange of trips between them. In other words, the truck does a mutual exchange giving a part of its responsibilities (loads) to another truck and receiving a part of the responsibilities (loads) from the exchanging truck in return. Similar to the transfer operator, the interchanging operator can be used to perform both intra exchange and inter exchange of trucks. This is also used to reduce the total unloaded miles as well as the total waiting time. While interchanging the trips, the first mill of the loaded miles is left undisturbed since that destination cannot be changed.

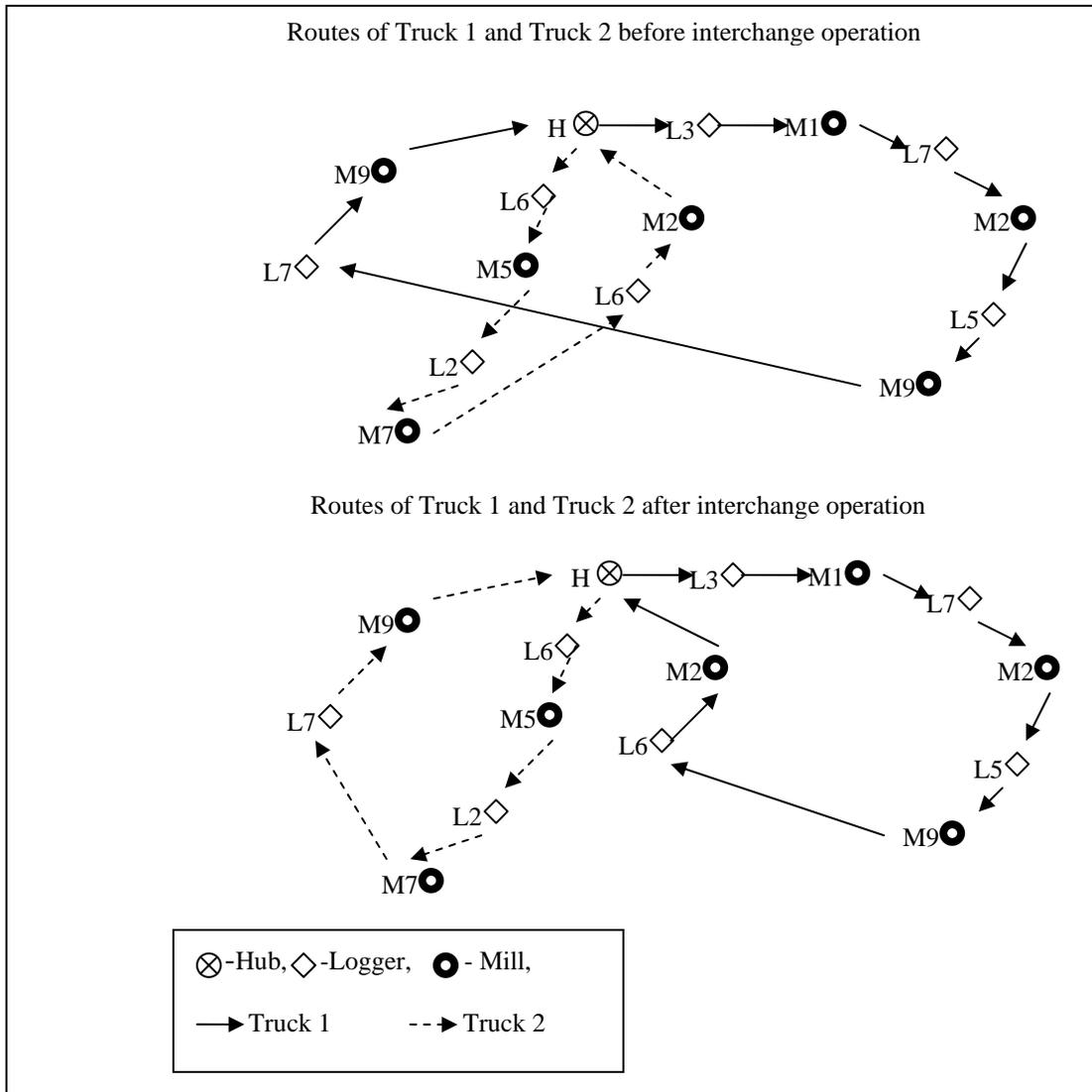


Figure 4.6: Interchange operator being applied to trucks' routes.

Figure 4.6 shows the routes of trucks 1 and 2 before and after the interchange operation. Logger 6 and Mill 2 of truck 1 are interchanged with logger 7 and mill 9 of truck 2 during the interchange operation. The interchange operator and the transfer operator help in finding the shortest routes for the trucks thereby optimizing the routes without compromising the feasibility of the solution.

The above operators when applied to the previous solution generate a new solution. The fitness value of the previous solution is known and that of the new solution is calculated.

F1 - Fitness value of the previous solution

F2 - Fitness value of the new solution

$$\delta = F1 - F2$$

If $\delta > 0$, then the new solution is adopted as the lowest search solution

Even if $\delta < 0$, the new solution is adopted as the lowest search solution with a probability that depends on the current temperature and the difference in the fitness value (δ).

4.3.4 CONDITION FOR ACCEPTING A BAD SOLUTION:

The condition for accepting a bad solution is given by

$$\text{Random}(0, 1) < (T + \delta)/T$$

If the above condition is not satisfied, then the bad solution is rejected and the current solution is adopted as the lowest search. Meanwhile the solution with lowest fitness found so far is kept in memory as the best solution. Every time a new solution is found, its fitness value is compared with that of the best solution apart from comparing it with the lowest search fitness. At each temperature, the neighborhood is searched for a fixed number of iterations to get the new solution.

Best solution - Solution which has the lowest fitness all through the annealing process

Lowest search - Current solution whose neighborhood is searched for improvement

New solution - The solution found in the neighborhood of the lowest search by applying the search operator to the lowest search solution.

After searching the neighborhood for a fixed number of iterations, the temperature is decremented by multiplying the temperature with a factor β . The factor β is fixed and remains constant throughout the process. The above process is repeated until the temperature reaches the final temperature value. During the initial few steps of the temperature, simulated annealing does not get stuck to the local optima and moves freely around all neighborhoods to get a good solution. As the temperature reaches the end, the solution gets stuck in the local optima.

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 fewer unloaded miles, less waiting time and less time duration to complete the trips. The best solution obtained is very close to the global optimum solution. The time taken to find a good solution is less than 1 min when simulated annealing is used whereas it takes days to find a good solution using a standard solver.

The set of routes 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.

In the beginning, a small problem is tried using integer programming. The constraints of the problem are mathematically modeled and the objective function is

formulated. It is then solved using the branch and bound algorithm programmed using Excel software. The results are tabulated and value of the objective function is noted. The small problem is described as follows.

4.4 DESCRIPTION OF THE SMALL PROBLEM:

T- Number of trucks = 4

M-Number of mills = 2

L-Number of loggers = 3

N-Number of trips = 4

Distance Matrix = {70, 67,
65, 76,
45, 51}

Demand Matrix = {1, 2,
2, 3,
2, 1}

In this problem, all the trucks are unloaded and our objective is to minimize the total unloaded miles traveled by all the trucks.

4.4.1 SOLUTION USING INTEGER PROGRAMMING MODEL:

The above description of the number of trucks, trips, mills and loggers are fed as an input to the programming model. Also the demand matrix and the distance matrix are defined into the system. The model is solved using the branch and bound algorithm and gives the set of routes for all the trucks and the total unloaded miles as its output as shown in Table 4.1

Table 4.1: 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 |

Total unloaded miles = 399

4.4.2 SOLUTION USING SIMULATED ANNEALING ALGORITHM:

The above problem is now solved using simulated annealing heuristic algorithm and the results are compared for the degree of closeness of the heuristic method with that of the integer programming model. The simulated annealing algorithm is integrated with the simulator. Hence the best set of routes found using the algorithm is given as an input to the simulator to find the total unloaded miles traveled by the truck as shown in table 4.2. It is found that the value of the objective function is same for the two models. The total unloaded miles found using simulated annealing algorithm is 399 miles and it matched exactly with that obtained using the integer programming model.

Table 4.2: Solution using simulated annealing 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 |

Total Unloaded Miles =399

By comparing the two solutions obtained using the integer programming model and the simulated annealing algorithm, it can be found that though the value of the

objective function is the same, the set of routes for the trucks is different. The problem contains more than one optimal solution.

In the case of small problems, the solution is obtained within a fraction of a second using the heuristics algorithm and obtained in less than 5 seconds using the branch and bound algorithm in the integer programming model. But in the case of large problems, it is possible to obtain the solution within 1 min using S.A. whereas it takes hours to solve using the standard algorithms available with integer programming model.

CHAPTER 5

DETERMINISTIC SIMULATOR

A simulator collects information regarding a truck's routing based on the input given to it. It is a program which emulates the movement of trucks during the day. The simulator keeps track of all the performance metrics as the truck takes its route through hub, loggers and mills. The simulation carried out here is *deterministic* since the events are derived from the routes of the trucks which are known already and given as input.

5.1 BASIC MODEL OF SIMULATOR:

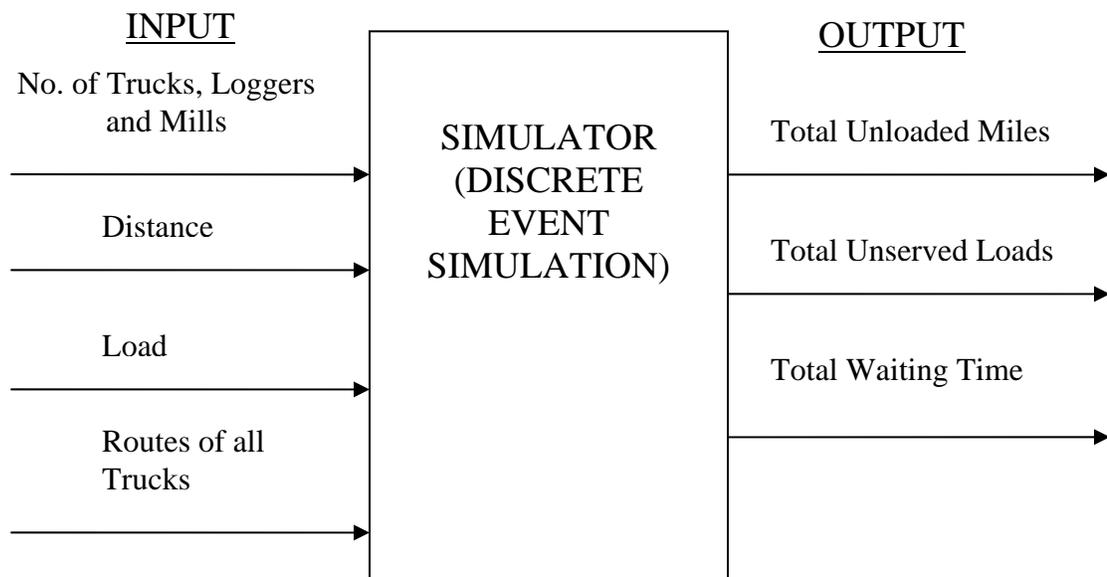
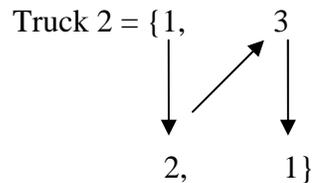


Figure 5.1: Basic model of a simulator.

Figure 5.1 above shows the basic model of the simulator. The total number of trucks used and number of loggers and mills are fed to the simulator as standard input along with the distance matrix and load requirement matrix. Also the routes followed by each truck on any given day should be entered into the simulator. The simulator emulates the movement of the trucks according to the routes given and gives the output in the form of total unloaded miles of the truck, total unserved loads and total waiting time of all the trucks in the various customer locations. For example, any truck's route can be defined as follows,



The truck starting from the hub goes to a series of customer locations like logger 1, mill 2, logger 3, and mill 1 and finally ends the day by returning to the hub. The simulator keeps track of the total unloaded miles as the truck moves from mill to logger. Similarly whenever the truck proceeds from logger to mill, the total loaded miles is incremented by the distance between the logger and mill. So at the end of the day, after emulating the movement of the trucks, statistics of total loaded miles and total unloaded miles are obtained. The mills and loggers are the customers for the trucks. The cranes are the servers at the customer locations. The working of a simulator is shown using a truck's route in figure 5.2.

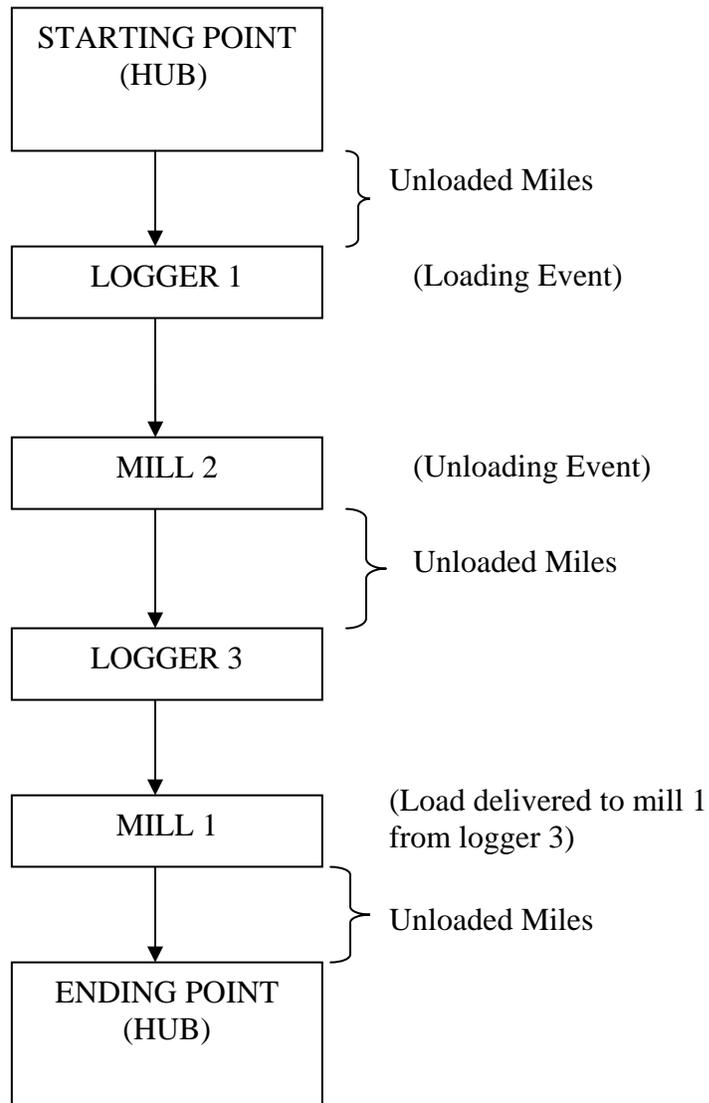


Figure 5.2: A simulator following a truck’s route.

Whenever there is a queue at the customer location, the truck has to wait before getting served by the cranes at logger and mill locations. Hence the total waiting time is calculated as the sum of the waiting times of all the individual trucks at all customer locations.

The truck delivers the loads from the logger to the mill. As the loads are delivered, the served load matrix is incremented by one every time. At last, the served load matrix is compared with the load requirement matrix. The total difference between the load requirement matrix and the served load matrix gives the unserved loads at the end of the day.

5.2 INITIALIZATION OF PERFORMANCE PARAMETERS:

All the statistics are initialized to zero during the beginning of the simulation. The trucks can start the day as loaded or unloaded which depends on the previous day's ending condition. Hence the starting condition of the trucks is given as an input in the beginning.

Discrete event simulation is used to emulate the movement of trucks. The departure and arrival from hub, mill and logger are considered as events. The events are enlisted in the main linked list according to the time at which the event must be emulated. There is time associated with the loading and unloading event. The simulator will first enlist an initial set of events in the list at time zero and starts executing it one by one. The clock is incremented by the time duration of the event taking place.

At times, the server may be busy serving the truck that has already arrived. The new trucks which have just arrived should be put in the queue. Hence a separate queuing list is created to keep track of the trucks waiting in the queue. Also the status of the server is changed from busy to idle whenever a truck leaves a customer location. First in first out (FIFO) strategy is implemented during the queuing process.

The time of the event that is executed currently is adopted as the current updated time. Since the events are inserted in the list along with the time at which they are executed, it becomes easier to adopt an event's execution time as the current time. The main list and queuing lists are filled with null character to indicate the end of the list. The servers at the logger and mill locations are set to idle status in the beginning with no queue in front of them.

5.3 SIMULATION OF THE EVENTS:

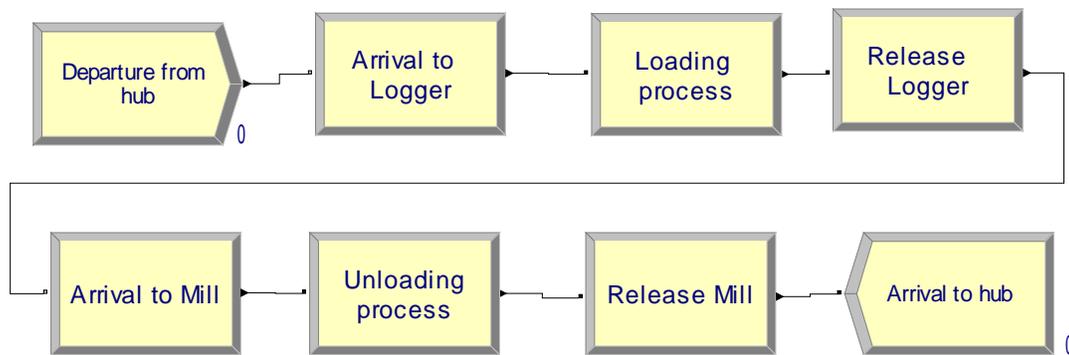


Figure 5.3: Schematic diagram of the simulator.

The simulation starts at time zero. At that time the served load matrix between the loggers and mills is set to zero. The total distance traveled, loaded miles and unloaded miles are set to zero in the beginning. An event namely an arrival or a departure event consists of truck number, logger number, mill number, trip number and the time. For all the trucks, the first event is the departure of the truck from the hub.

The next destination for any truck is obtained from the set of routes given as the input. A schematic diagram of the simulator is shown in figure 5.3. The destination may be a logger if the truck is an unloaded truck and would be a mill if the truck is loaded. The destination along with the arrival time and truck number is inserted as an arrival

event in the main event list. Arrival time at the destination is calculated based on the current time plus the traveling time calculated based on the distance between the hub and the destination divided by the speed.

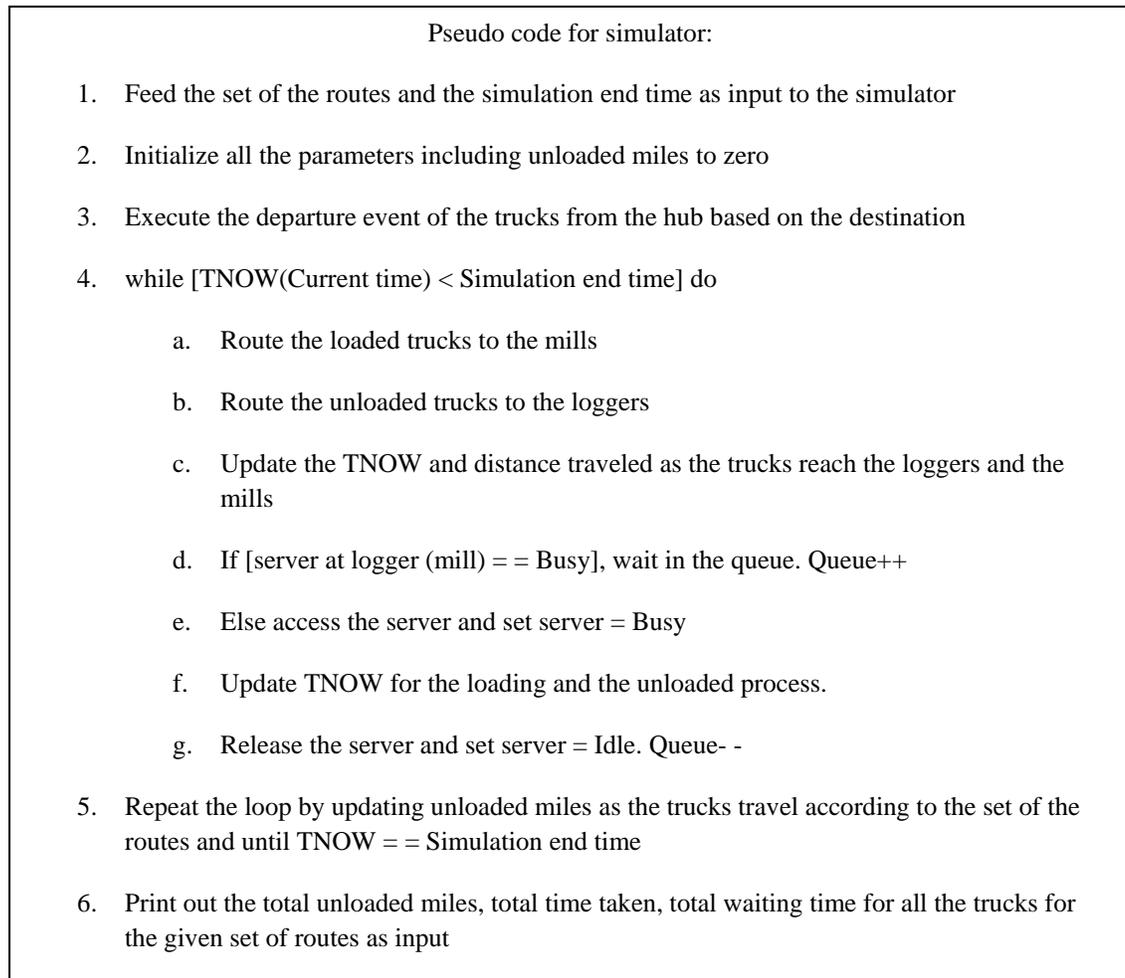


Figure 5.4: Pseudo code for working of simulator.

At time zero, all the trucks are dispatched to the corresponding destinations. As soon as the truck reaches a logger or a mill, the status of the server at the destination is checked for idleness. In the beginning of the day, the servers will be idle until a truck approaches it for service. The first truck to reach the destination gets direct access to the

server without waiting in the queue and is served immediately with a service time of 6 min for loading or unloading process. When the truck is being served by the server, the status of the server is set to busy. All the other trucks which arrive when the server is busy will be put in a queue based on a first in first out (FIFO) strategy.

The queuing time is the difference between the time at which the server starts serving the truck and the time at which the truck arrives at the location. The simulator sums up the waiting time of all the trucks in the system throughout the day.

The status of the server will be switched back to idle after serving a truck. The first truck from the queue is pulled up from the queuing event list and will be served changing the status of the server to busy again. This process is repeated throughout the day. There is time associated with the loading and the unloading process both at the logger and the mill locations. The duration of the loading and unloading process are assumed to be fixed.

The arrival time is incremented by the sum of the waiting time and the service time and inserted as an event for the release of the server to serve other trucks. Upon releasing the server, the next destination for the truck is inserted as an event along with the time the truck takes to reach the destination. The loaded miles are updated when the truck reaches the mill, and similarly, the unloaded miles are updated when the truck reaches the logger.

While releasing the server, the trip is incremented by one when the truck has visited an equal number of the loggers and the mills. Depending on the distance traveled, the trucks may vary in the number of trips taken to cover the distance. During the last trip

of all the trucks, the trucks should be forced to return to the hub. Hence the last destination for all the trucks is the hub. The destination is checked every time the truck arrives to a mill or a logger. If the destination is the hub, the truck's movement is stopped and all the statistics like the miles traveled and the waiting time are calculated for that particular truck.

The simulation end time, the time until which the simulation should be executed, is given as an input in the beginning. As soon as the current time reaches the simulation end time, the simulation is stopped and all the remaining events are listed out. If there are no events remaining, it indicates that the trucks have completed all the trips in the route assigned to it. On the other hand, if there are any events remaining, it means that the trucks take more time to serve the loads. At last, all the statistics namely the unmet loads, total loaded miles, total unloaded miles and the total waiting time are printed out.

5.4 ADVANTAGES AND USES OF THE SIMULATOR:

The simulator is a useful tool to track the movement of any single truck throughout its route as it moves from a logger to a mill and vice versa. Given any single truck's ID as input, the simulator prints out the sequence of events the truck is following along with the time spent at various locations.

The potential of the simulator lies in the successful execution of the events in the main event list, thereby tracking the movement of the trucks from one location along with the simultaneous collection of the distance related and the time related statistics.

The output from the simulator is analyzed to find the proportion of the total loaded miles to the total miles traveled. A good set of routes always results in the total

loaded miles greater than the total unloaded miles. Also, the total time wasted by all the trucks in queuing is calculated by the simulator. These statistics are used as a mode of comparison with the statistics obtained using a different set of routes.

Apart from the regular output of statistics, the simulator also gives information about the trucks which take more time to complete the trips than the simulation end time. For each truck, there is an end time at which it reaches the hub after serving all the loads. The end time for all the trucks is given as an output with which we can sort out the trucks which take more time to complete the trips apart from the trucks which takes less time to complete the trips.

Even if a different set of the routes is given as an input, the simulator recalculates the statistics and gives a different output. Hence the simulator can be used any number of times to calculate the statistics for any set of input given to it. A fitness function is defined at the end of simulation which allocates a penalty to each unloaded mile, each unmet load and each hour of waiting time and calculates a value corresponding to the set of routes given as an input. The value of the fitness function determines the efficiency of the set of the routes.

The main advantage of the simulator is that it is very fast and returns the output within a fraction of a second. Also it is very robust and can be used to simulate routes consisting of a large number of trucks, trips and the customer locations.

CHAPTER 6

EXPERIMENTAL RESULTS OF ACTUAL SYSTEM

The existing log truck transportation in the states of Louisiana and Mississippi states is quite big compared to the known problem discussed while comparing the efficiency of the simulated annealing algorithm with that of the integer program. The data regarding the existing system is provided by the Department of Forestry in an Excel file. The recording system tracks the movement of the trucks as they move between the hub, loggers and mills and records the data along with the duration of the stoppage, distance traveled, time taken, loading time and the unloading time etc.

The data is well analyzed to find the load requirement matrix, distance matrix between loggers and mills. Also the number of trucks, loggers, mills and trips are extracted along with the set of routes taken by each truck during the delivery of the load all through the day. The set of routes is then fed to the simulator to get the total unloaded miles, total waiting time, total time duration to serve the loads etc. It is necessary to obtain these performance metrics to evaluate the fitness function and also to compare it with the performance metrics of the improved system. The problem can be described as follows.

6.1 DESCRIPTION OF THE ACTUAL PROBLEM:

T- Number of trucks = 68

M-Number of mills = 13

L-Number of loggers = 22

N-Number of trips = 7

Loaded trucks = 10

Unloaded trucks = 58

The number of trips taken to deliver the loads is found to be seven for the existing system but in order to complete the trips within an 8 hour time duration, the number of trips is reduced to five to get a better solution.

The above parameters along with the demand matrix and distance matrix are given as an input to the simulated annealing program to get the best set of routes that can be followed by the trucks in delivering the loads. The output from the simulated annealing algorithm combined with the simulator is compared against the performance metrics in the existing system.

6.2 FINISHING TIME OF TRUCKS:

The finishing time of trucks is an efficient metric to analyze the utilization of the trucks. Also it is an estimate to find whether all the trucks have finished within the time window of 8 hours.

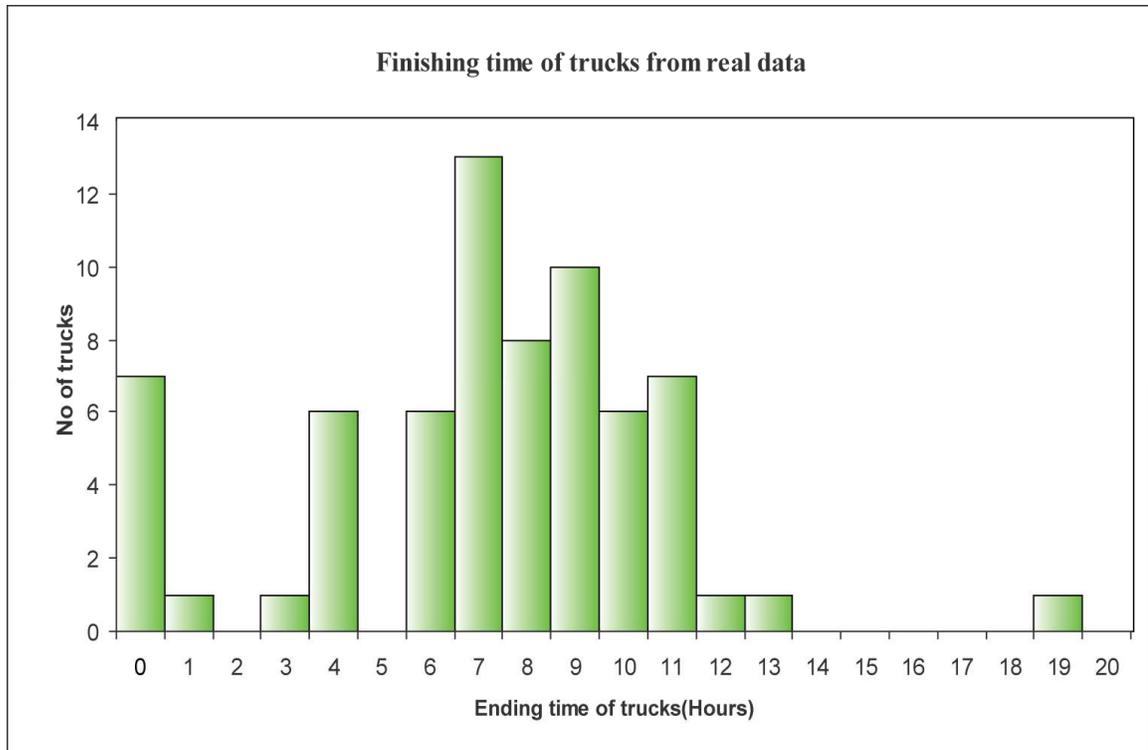


Figure 6.1: Finishing time of trucks in the existing system.

The figure 6.1 shows the finishing time of trucks in the existing system obtained from simulator by giving the current set of routes as input. The mean finishing time of trucks is 7.21 hours with standard deviation of 3.61 hours. It is observed that the last truck goes to the hub only after 18 hours. There are about 34 trucks which take more than 8 hours to deliver the loads. Also there are about 7 trucks which are not used to deliver the loads.

It can be understood that the existing system is very inefficient since it takes more time than the shift duration to deliver the loads; therefore, drivers, crane operators at the loggers and mills have to be paid more. All the trucks are not utilized properly. Some of the trucks are underutilized and some are over utilized.

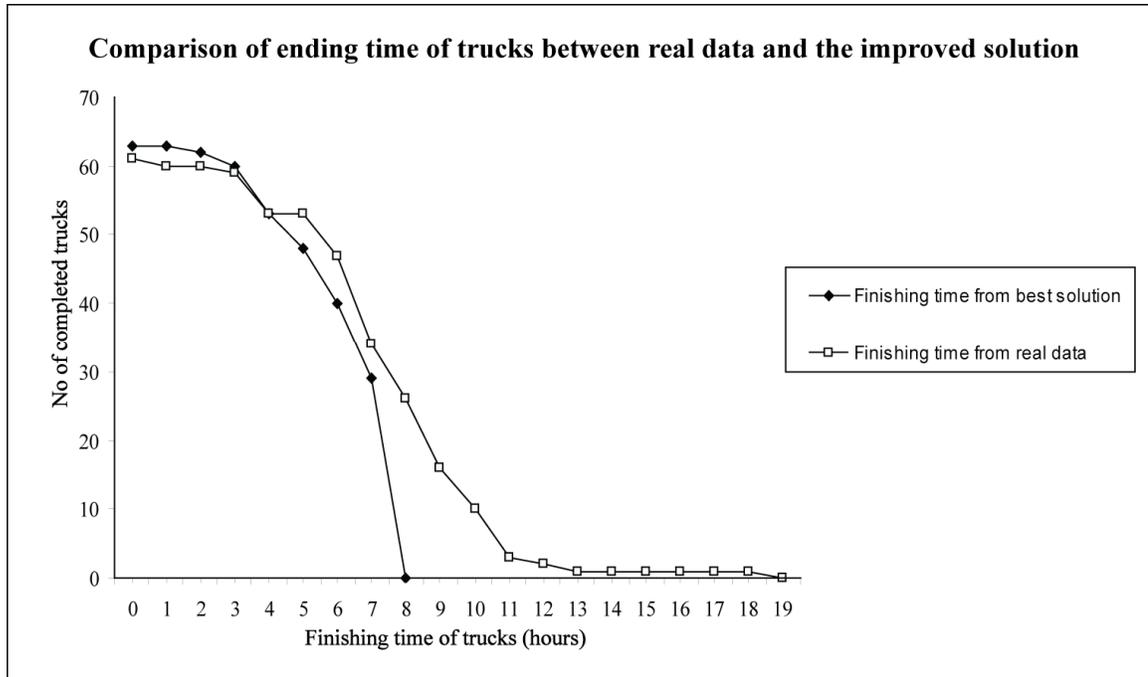


Figure 6.2: Comparison of finishing time of trucks between real data and the best solution.

The figure 6.2 shows the ending time of trucks in the improved solution and in the real data. From the figure it is evident that all the trucks finish the job within 8 hours when compared to the ending time of 19 hours for trucks in the existing system from real data. The mean finishing time of trucks is 5.76 hours with standard deviation of 2.28 hours. The standard deviation is very small when compared to the existing system and the majority of the trucks are effectively utilized. But there are still 28 trucks finishing the day within 6 hours but 5 trucks are not used for the day to serve the loads and stay at the hub throughout the day.

6.3 IMPROVEMENT IN PERFORMANCE METRICS:

The first and foremost aim is to reduce the unloaded miles traveled by all the trucks. Also the trucks should deliver all the loads within the time duration of 8 hours

with less waiting time at the logger and mill locations. It is necessary to compare the performance metrics of the existing system with that of the best solution obtained using simulated annealing to calculate the improvement percentage.

Table 6.1: Improvement in performance metrics.

| | Existing system | Optimal system | Improvement (%) |
|----------------------------|-----------------|----------------|-----------------|
| Total Miles | 20630 | 17127 | 17.00 |
| Loaded Miles | 10028 | 9509 | 5.17 |
| Unloaded Miles | 10602 | 7618 | 28.14 |
| % Loaded Miles(L.M/T.M) | 48.60 | 55.54 | 14.28 |
| % Unloaded Miles | 51.39 | 44.48 | 13.72 |
| Total Waiting time (hours) | 21.00 | 13.88 | 33.90 |

a) TOTAL DISTANCE TRAVELED:

It is found that the total distance traveled by the trucks is reduced from 20,630 miles to 17,127 miles. The percentage of improvement is 17%. Since the total distance traveled is reduced, the trucks are able to meet the schedule within the 8 hour duration. It saves overtime pay for the truck owners, logger owners and mill owners.

b) LOADED MILES TRAVELED:

The loaded miles in the existing system is 10,028 miles which was improved by 5.17% to 9,509 miles in the newly found best solution using S.A. There is almost no scope 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 is due to the fact that there is

a difference in the number of trucks that end the day as loaded in the hub. There are many fewer trucks than in the existing system which end the day as loaded. Hence there is a small improvement in the loaded miles percentage.

c) UNLOADED MILES TRAVELED:

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 so that it can reduce the total unloaded distance traveled. The improvement heuristic plays an important role in improving the solution by reducing the unloaded miles. The solution has shown a tremendous improvement by reducing the total unloaded miles from 10,602 to 7,618 with an improvement percentage of 28.14%. 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.

d) RATIO OF LOADED MILES TO TOTAL MILES:

The ratio of loaded miles to total miles is known as loaded miles percentage. This is an industry standard to measure the efficiency of the proposed system. The loaded miles percentage improved from 49% to 56% showing an improvement of 14.28%. It is considered to be a poor model whenever the loaded miles percentage is less than 50%. Since in the solution obtained, the loaded miles percentage is 56%, it is said to be a good model.

e) TOTAL WAITING TIME:

The total waiting time of all the trucks is reduced from 21 hours to 13.88 hours. In other words, on an average the waiting time of trucks has been reduced from 18.53 minutes per truck to 12.25 minutes per truck with the percentage of improvement being 34%.

Table 6.2: 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 |

The average waiting time of trucks at each logger location is given in Table 6.2. The trucks have to wait more at logger 4, where the average waiting time is 10.71 minutes since the logger 4 has 14 loads associated with it. Though there are only 6 loads supplied from the logger 1, the truck has to wait for 10 minutes on an average to get loaded. Trucks arrive at regular intervals at logger 19 and hence there is no waiting time even though there are 15 loads to be supplied from the logger.

Table 6.3: Average waiting time at mill locations.

| Mill No. | Total No. 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 |

The average waiting time at each mill location is shown in Table 6.3. 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 minutes and 6.80 minutes on an average since both the mills require a higher number of nearly 38 loads to be delivered it. At the same time, the trucks arrive at equal

intervals to the mill 3 and hence we have a lower waiting time of 3.36 minutes on an average.

CHAPTER 7

SENSITIVITY ANALYSIS

Sensitivity analysis is carried out to find the marginal effect of removing a truck, the effect of the distribution of loggers and mills and also the effect of location of individual logger and mill on the solution. Sensitivity analysis is done to find ways to reduce the cost associated with the operation and management of the log truck and thereby increasing the profit to the truck owners. It also helps in better designing of the locations of loggers and mills to meet the demand with less cost and time in the future.

7.1 REDUCTION OF TRUCKS:

There is huge savings associated with the reduction of trucks in the form of maintenance and rental savings per day for trucks not used and also in the form of wages for drivers.

The existing system consists of 68 trucks and looking into the best solution obtained from S.A as shown in figure 7.1, it is evident that all trucks finish the routing within 8 hours. Still there are 5 trucks which are not used to serve the loads. Also there are 15 trucks which work for less than 5 hours to serve the loads. Hence there is scope to reduce the number of trucks used and thereby to increase the utilization of the used trucks. It is necessary to check the effect of the reduction of the trucks on the total time taken to serve the loads, unloaded miles, time taken to find the optimal solution etc.

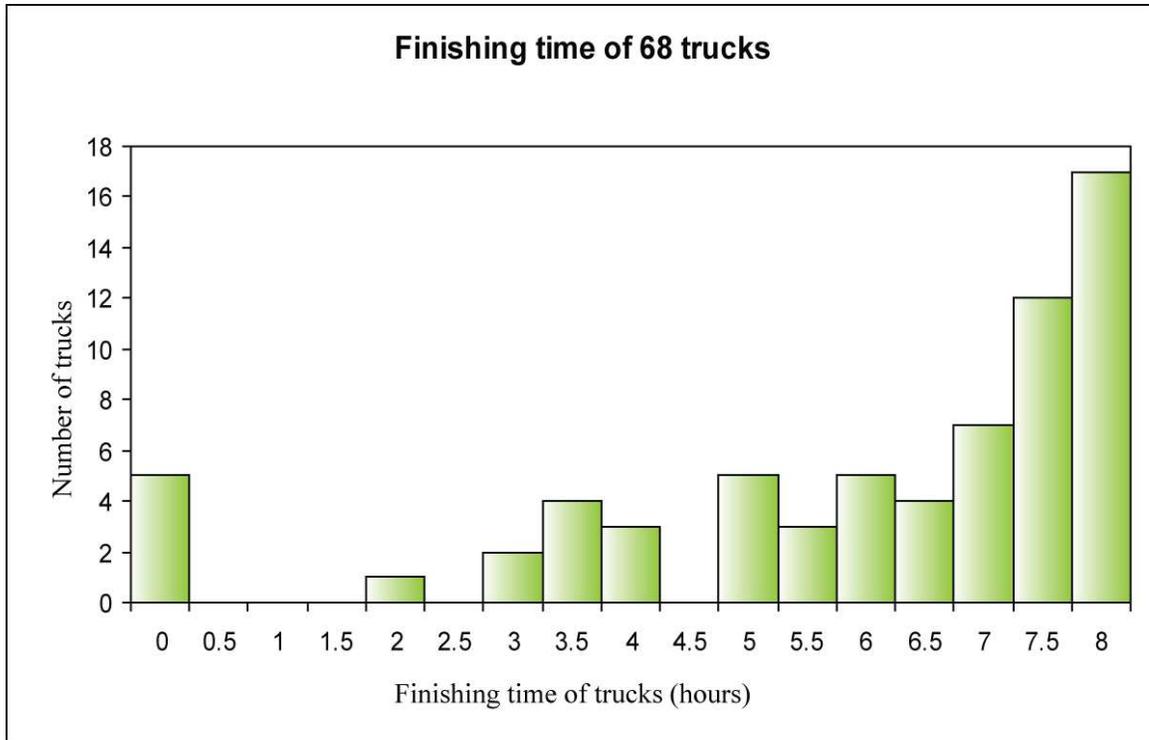


Figure 7.1: Finishing time of 68 trucks from the optimal solution.

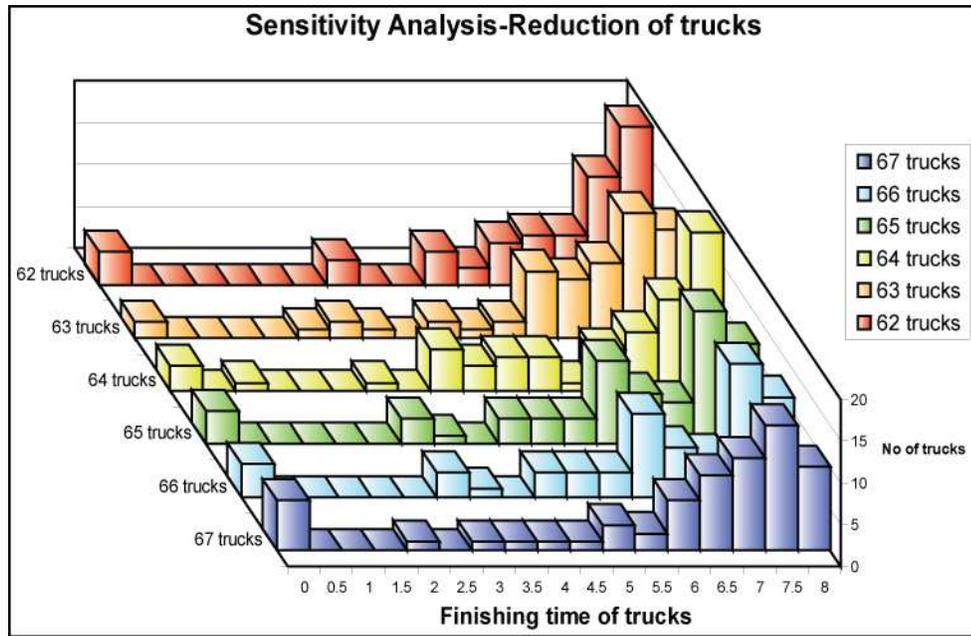


Figure 7.2: Sensitivity analysis for reduction of trucks from 67 trucks to 62 trucks.

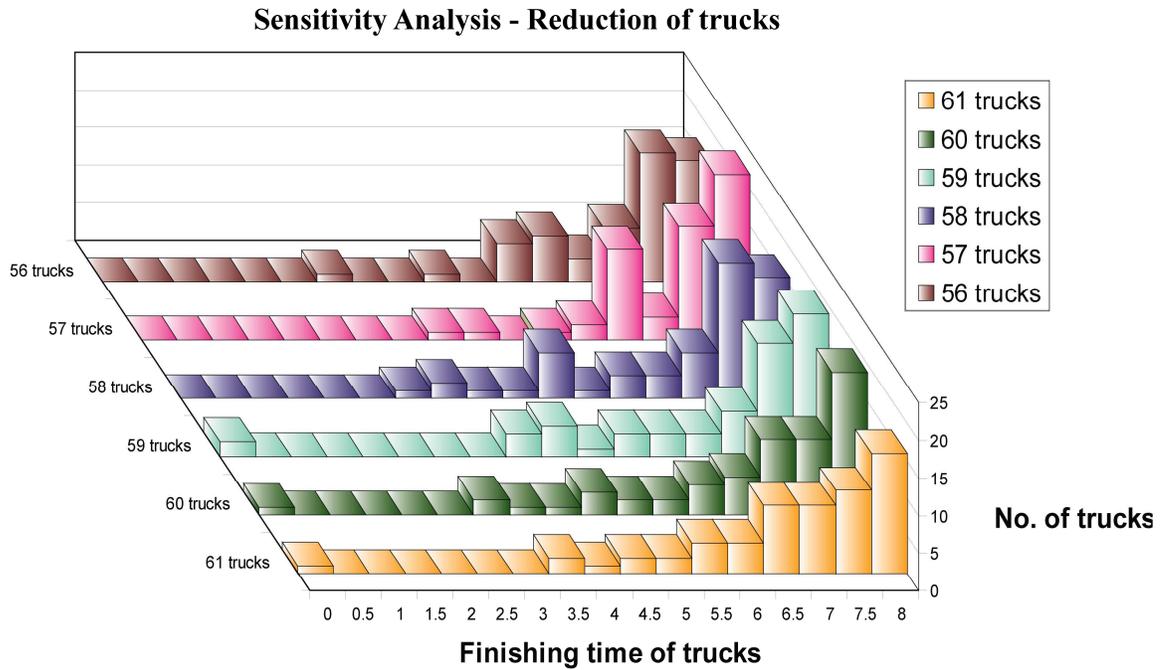


Figure 7.3: Sensitivity analysis for reduction of trucks from 62 trucks to 56 trucks.

The figures 7.2 and 7.3 show the sensitivity analysis for the reduction of trucks from 67 to 56 trucks. It is observed from the graph that as the trucks are reduced from 67 to 56, the finishing time of trucks is more concentrated towards the maximum of 8 hours. The higher the finishing time, maximum is the utilization of the trucks.

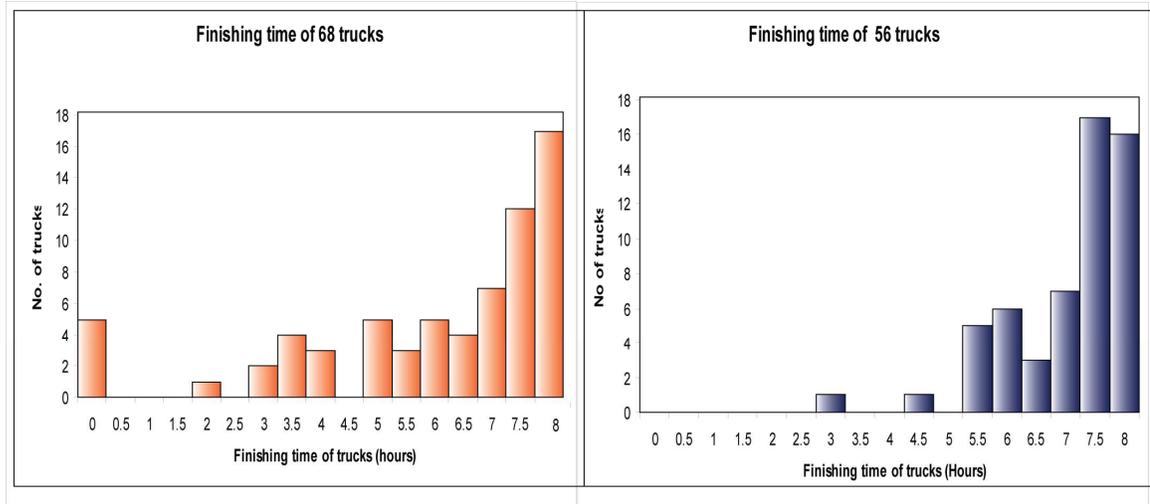


Figure 7.4: Finishing time of trucks for 68 trucks and 56 trucks.

When 56 trucks are used to serve the loads, it is found that no truck was left idle and except for 2 trucks and almost all trucks are active more than 5.5 hours. The figure 7.4 shows the finishing time of 68 trucks and 56 trucks. The total time of 8 hours is sufficient to meet the demand and there was no appreciable increase or decrease found in the percentage of unloaded miles. But as the trucks are reduced, the simulated annealing algorithm takes more time to find the solution for the problem.

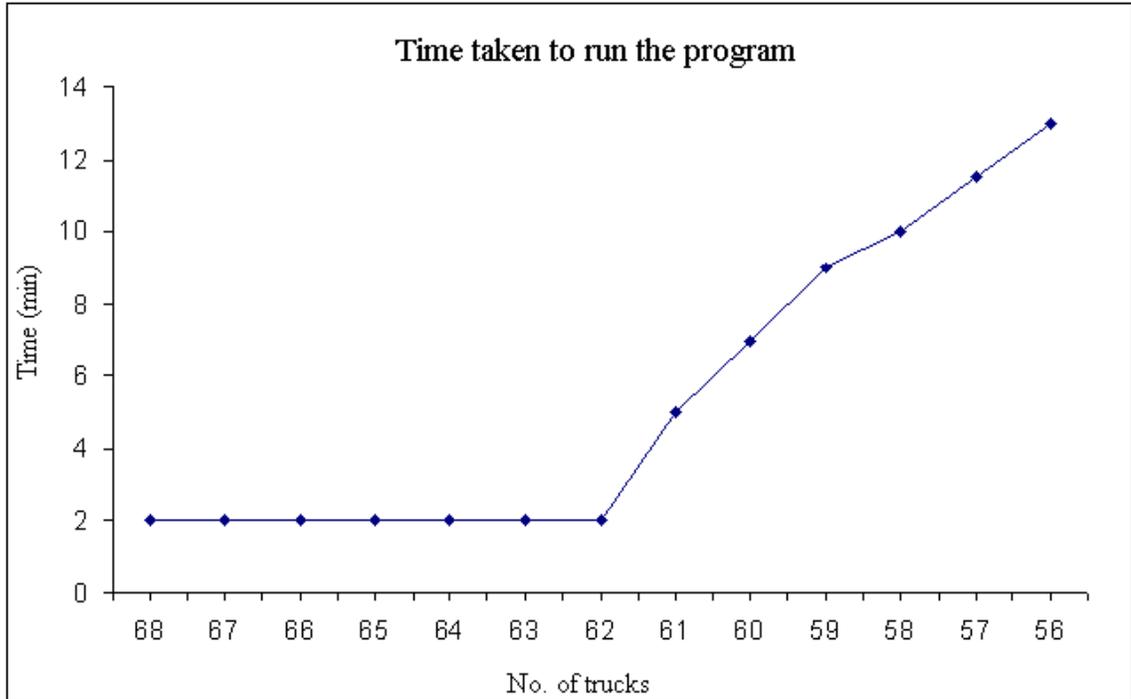


Figure 7.5: Time taken to find the solution with reduction in number of trucks.

The time taken to find the solution is approximately 2 min for 68 trucks through 62 trucks. This can be clearly seen in figure 7.5. As the trucks are further reduced, the relationship between the number of trucks and time taken to solve the problem was found to be inversely proportional and linear. The solving time increased from 2 min to 15 min for 56 trucks due to the fact that as the resources are reduced, the simulated annealing algorithm is taking more time to find a solution to the problem.

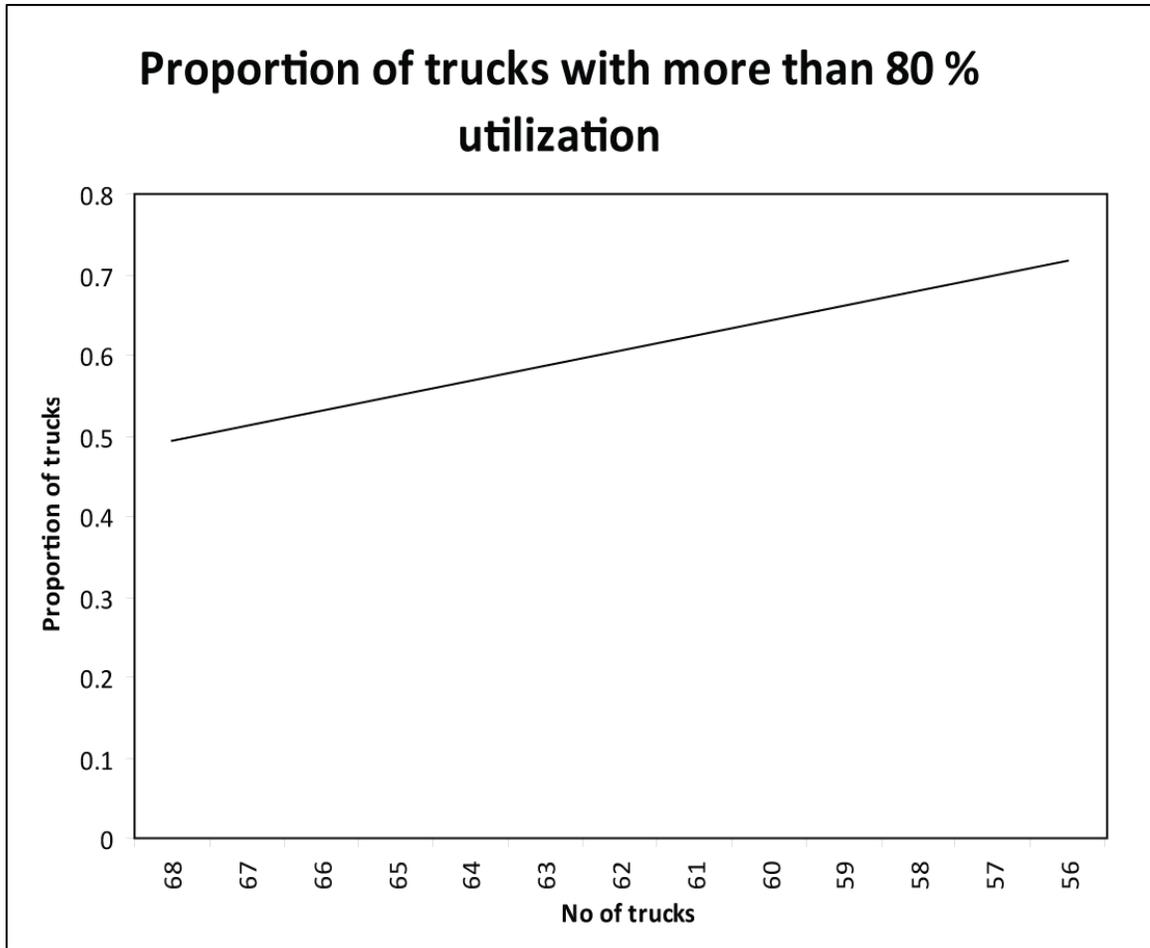


Figure 7.6: Proportion of trucks with more than 80% utilization.

When the system was run with 68 trucks, half of the trucks were utilized for more than 6.5 hours in 8 hours shift. This can be seen from the line of best fit plotted between proportion of trucks with more than 80% utilization and number of trucks shown in the figure 7.6. As the number of trucks is reduced, a larger proportion of trucks was utilized for more than 6.5 hrs. When the system was finally reduced to 56 trucks, more than 70% of the numbers of trucks were utilized for more than 6.5 hours.

7.2 SCENARIO ANALYSIS:

It is necessary to understand the distribution of loaded miles percentage in order to find the expected value of loaded miles percentage. The percentage of loaded miles varies with the geographical distribution of location of loggers, mill and hub. The distribution of loaded miles percentage gives us an estimate of the minimum value of loaded miles with a certain probability. The people working in the industry are more interested to find that value which can ensure at least a basic profit margin for the operating day.

Locations of all the loggers, mills and hub are marked on the map using Google Earth and an imaginary smallest possible rectangle is drawn over them so that its boundaries enclose the loggers, mills and the hub. The length and breadth of the rectangle is converted into distance in miles. Now the sides of the rectangle form the x-axis and y-axis for generating the locations of loggers and mills randomly. The length of the x-axis is 139 miles and that of y-axis is 110 miles. The loggers, mills and hub are located in a rectangular region by sampling the x-distance and y-distance.

A total of 175 scenarios are generated to extract the percentage of loaded miles for each scenario. The figure 7.7 shows six sample scenarios of uniform distribution of loggers, mills and hub. For each scenario, the loggers and mills are randomly located in the rectangular region. The distance between the loggers and mills is found by using the diagonal distance between the loggers and mills. For nearby locations, the diagonal distance is directly taken as the shortest distance of travel. In case of farther locations, the diagonal distance is multiplied by a factor up to 1.2 to obtain the shortest travel distance.

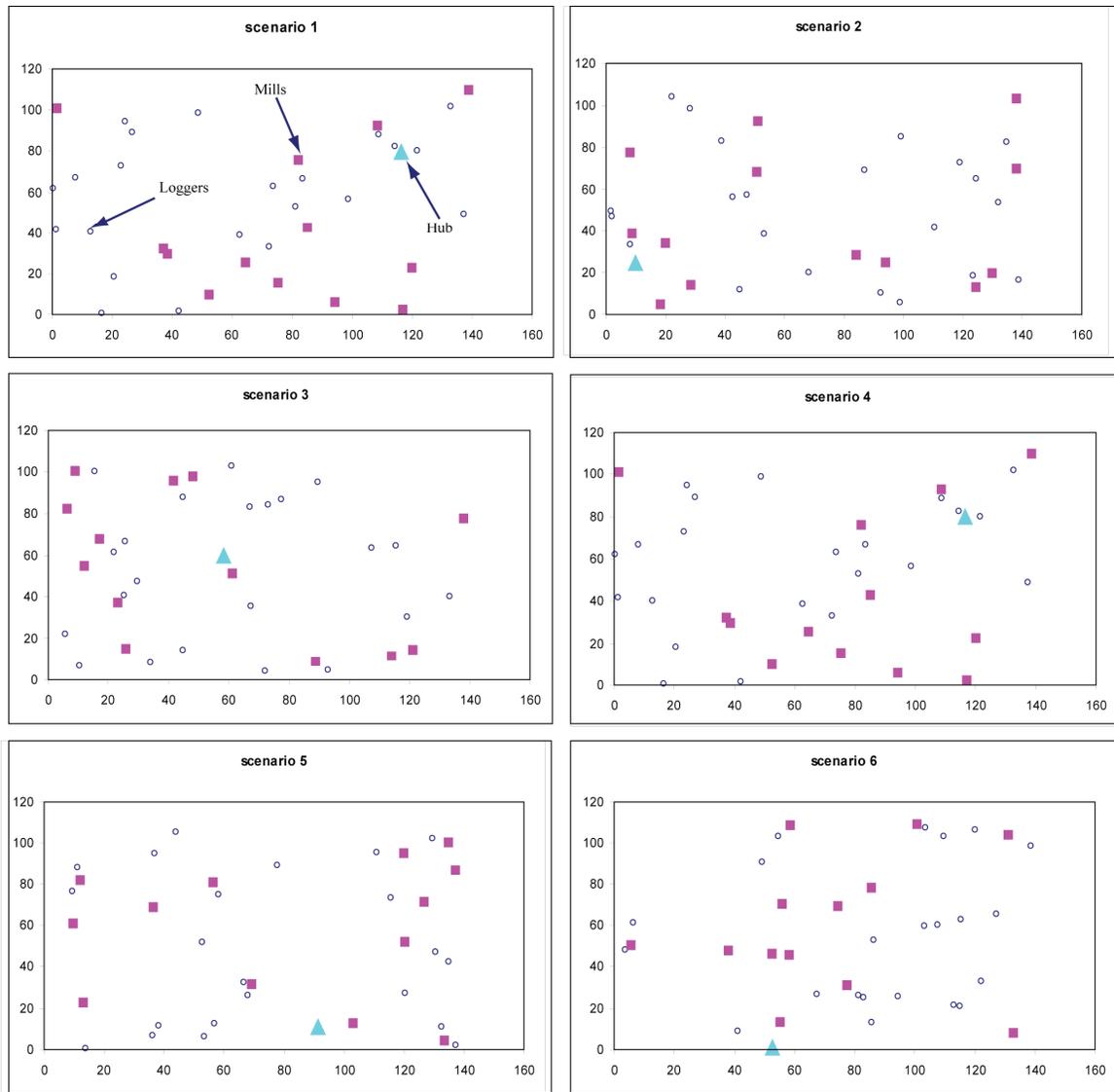


Figure 7.7: Six sample scenarios showing uniform distribution of loggers, mills and hub.

The distance matrix is different for each scenario and hence it is generated and fed each time to the simulated annealing algorithm. The simulated annealing algorithm tries to find an optimal solution for the scenario and the simulator calculates the percentage of loaded miles for the best solution generated by the simulated annealing algorithm. This

procedure is repeated for all the 175 scenarios and the percentage of loaded miles from each scenario is plotted to get the distribution of the loaded miles percentage.

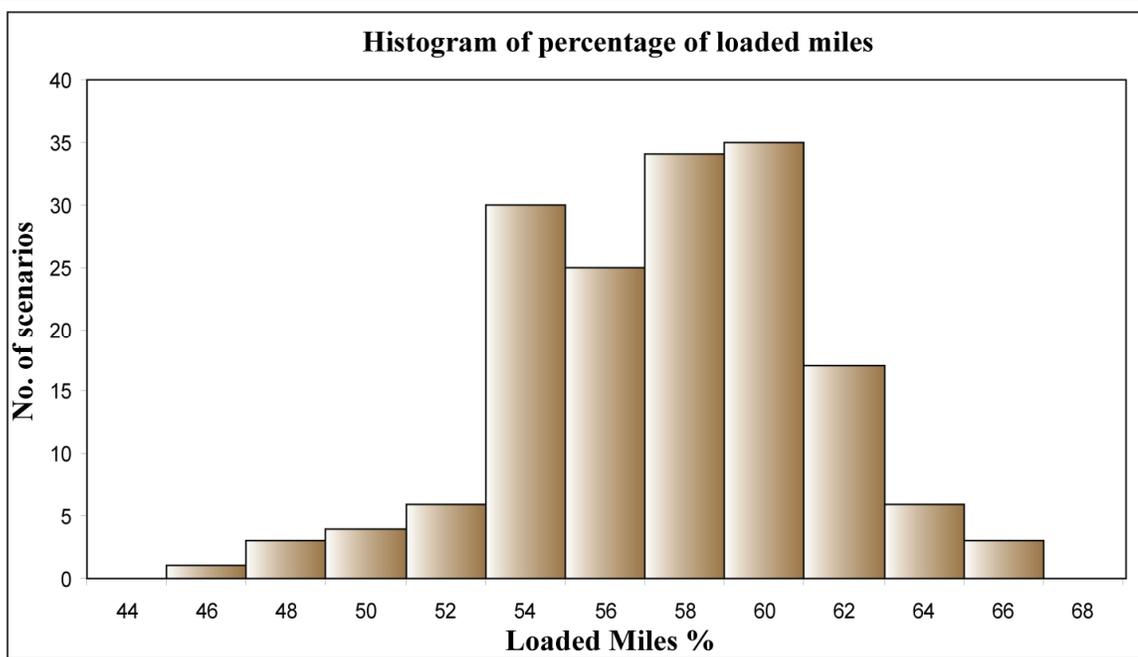


Figure 7.8: Histogram of loaded miles percentage for 175 scenarios.

The figure 7.8 shows the percentage of loaded miles for 175 scenarios. The mean loaded miles percentage is 57% with standard deviation of 3.7% and variance of 0.001. There were 4 scenarios where loaded miles were less than 50%. For these scenarios, the loggers and mills were located far away from each other. The main reason for the low percentage of loaded miles is that it spends most of the time in finding a feasible solution rather than improving it. For those cases, the number of searches at each temperature is increased so that the algorithm runs a little longer than 2 minutes to find a good solution.

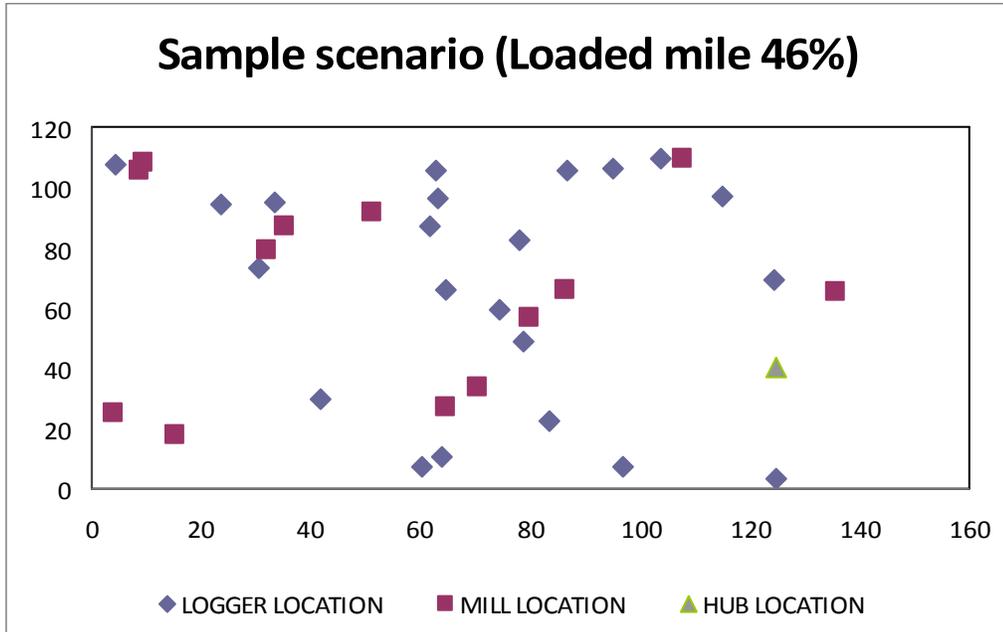


Figure 7.9: Sample scenario for loaded mile 46%.

The sample scenario where the model results in a low percentage of loaded miles is shown in Figure 7.9. The possible reasons being that the hub is located at one end of the rectangular region and also that the loggers are concentrated in the center of the region. Most of the loggers and mills are located far away from the hub. Since the trucks have to start the day from the hub, they travel more unloaded miles in reaching the logger in the beginning of the day. Similarly, after delivering the last load to the mill, the trucks have to travel more without carrying load to reach the hub.

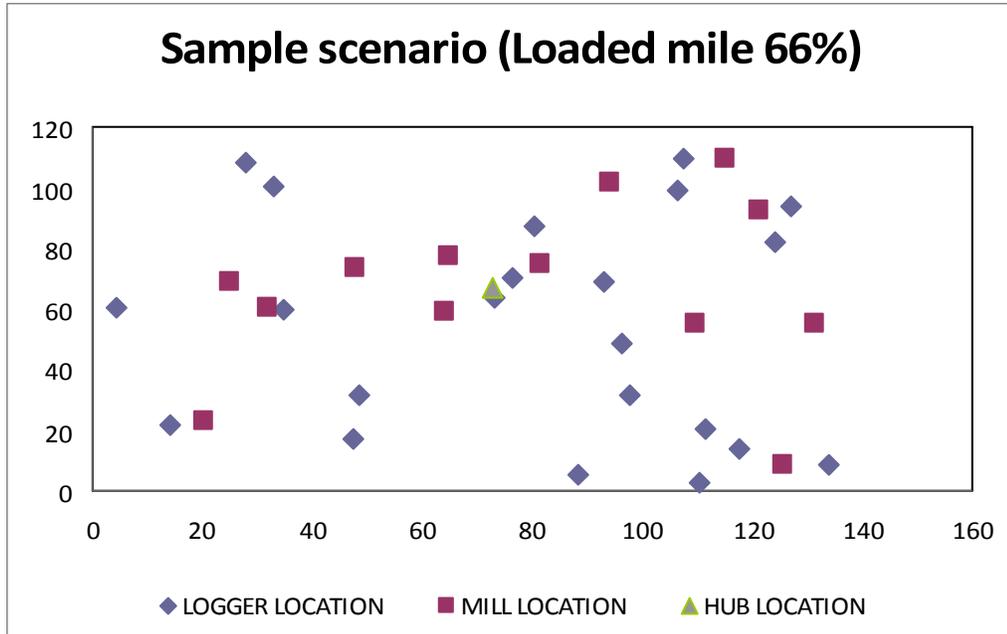


Figure 7.10: Sample scenario for loaded mile 66%.

The sample scenario where the model results in high percentage of loaded miles is shown in Figure 7.10. Since the hub is located in the center of the region, the trucks travel less unloaded miles in reaching the loggers from the hub in the beginning of the day. Similarly during the end of the day, the trucks travel less unloaded miles from the mills to the hub. Also, the loggers and mills are uniformly distributed over the whole region.

7.3 MARGINAL CONTRIBUTION OF LOGGERS AND MILLS:

The loggers, mills and the hub are located far away from each other. Each pair of loggers and mills has its own set of demand and is different when compared to other existing pair of loggers and mills. It is important to know the marginal contribution of every logger and mill on the entire system.

The marginal contribution of all the loggers and mills is calculated by removing one logger or one mill from the system every time. As the logger is removed from the

system, the demand matrix and distance matrix is updated to reflect the changes in the system. This new system is now fed to the simulated annealing algorithm to find the best possible set of routes for the trucks and the total unloaded miles.

The marginal contribution of the customer sites can be tabulated in terms of cost savings by comparing the ratio of unloaded miles to total loads between the old system and new system. Old system is one which has all the loggers and mills. New system is one in which one specific logger or mill is removed from the system. Cost is associated with the additional unloaded miles that the trucks have to travel to meet the demand associated with the logger or mill.

a) WITH LOGGER/MILL:

$$K1 = \frac{\textit{Unloaded miles}}{\textit{Total Loads}} (\$/\textit{Load})$$

b) WITHOUT LOGGER/MILL:

$$K2 = \frac{\textit{Unloaded miles}}{\textit{Total Loads}} (\$/\textit{Load})$$

c) COST SAVINGS:

$$\textit{Cost savings} = \frac{(K2 - K1)}{K2} (\%)$$

A logger or mill is said to be beneficial to the system when the trucks travel more unloaded miles to meet the demand without that particular logger or mill. In other words, the ratio of K2 should be greater than that of K1 for a logger to be beneficial.

Table 7.1: Results of marginal contribution of loggers.

| | With Logger K1=Unloaded miles/ total loads (\$/Load) | Without Logger K2=Unloaded miles/ total loads (\$/load) | Cost Savings (K2-K1)/K2 (%) |
|--------|---|--|-----------------------------------|
| Log 1 | 44.29 | 44.70 | 0.91 |
| Log 2 | 44.29 | 45.16 | 1.92 |
| Log 3 | 44.29 | 49.60 | 10.71 |
| Log 4 | 44.29 | 43.03 | -2.93 |
| Log 5 | 44.29 | 45.22 | 2.06 |
| Log 6 | 44.29 | 44.38 | 0.21 |
| Log 7 | 44.29 | 44.32 | 0.07 |
| Log 8 | 44.29 | 44.01 | -0.63 |
| Log 9 | 44.29 | 43.27 | -2.36 |
| Log 10 | 44.29 | 45.16 | 1.93 |
| Log 11 | 44.29 | 43.12 | -2.72 |
| Log 12 | 44.29 | 42.10 | -5.19 |
| Log 13 | 44.29 | 47.38 | 6.52 |
| Log 14 | 44.29 | 45.11 | 1.82 |
| Log 15 | 44.29 | 47.74 | 7.23 |
| Log 16 | 44.29 | 44.38 | 0.20 |
| Log 17 | 44.29 | 42.75 | -3.60 |
| Log 18 | 44.29 | 44.74 | 1.01 |
| Log 19 | 44.29 | 46.84 | 5.44 |
| Log 20 | 44.29 | 43.99 | -0.68 |
| Log 21 | 44.29 | 44.18 | -0.26 |
| Log 22 | 44.29 | 45.85 | 3.41 |

Results of marginal contribution of all the loggers are shown in Table 7.1. Based on the cost savings, logger 3 is found to be the most beneficial logger since the

percentage of cost savings in 10.71. A few of the other beneficial loggers are logger 13, logger 15 and logger 19. Logger 12 and logger 17 increases the cost associated with meeting the demand since they have negative cost savings.

Table 7.2: Results of Marginal contribution of mills.

| | With Mill K1=Unloaded miles/ total loads (\$/Load) | Without mill K2=Unloaded miles/ total loads (\$/Load) | Cost Savings (K2-K1)/K2 (%) |
|---------|---|--|-----------------------------------|
| Mill 1 | 44.29 | 55.25 | 19.84 |
| Mill 2 | 44.29 | 52.99 | 16.41 |
| Mill 3 | 44.29 | 47.66 | 7.08 |
| Mill 4 | 44.29 | 45.32 | 2.27 |
| Mill 5 | 44.29 | 47.22 | 6.21 |
| Mill 6 | 44.29 | 44.80 | 1.13 |
| Mill 7 | 44.29 | 40.24 | -10.06 |
| Mill 8 | 44.29 | 44.85 | 1.24 |
| Mill 9 | 44.29 | 44.07 | -0.50 |
| Mill 10 | 44.29 | 44.11 | -0.42 |
| Mill 11 | 44.29 | 45.40 | 2.45 |
| Mill 12 | 44.29 | 44.69 | 0.89 |
| Mill 13 | 44.29 | 44.29 | 0.00 |

Similarly, the marginal contribution of mills is done by removing one mill at a time from the system. The marginal contribution of mills is shown in Table 7.2. Mill 1 and Mill 2 are few of the beneficial mills since they yield a positive cost savings of 19.84 and 16.41 respectively when included in the system. Mill 7, Mill9 and Mill 10 gave negative cost savings and hence by including them, the ratio of unloaded miles per unit load for serving the demand increases.

In the future, if there is a need to include more loggers or mills into the system, profitability and feasibility of the system can be found by doing a marginal analysis for the new mills and loggers. If the cost savings turns out positive for a particular customer site, then that beneficial logger or mill can be included to increase profit.

CHAPTER 8

CONCLUSION AND DISCUSSION

The goal of this research was to develop a system which can produce a solution with less time than the integer programming approach but very close to the optimal solution for the log truck scheduling problem with capacity and time window constraints. A system was developed using simulated annealing algorithm to solve the problem with solving time of less than 2 minutes and a solution much better than the existing one is obtained. The system was developed using C program and solved using a computer with Intel Pentium 4 processor having 2.40 GHz of processing speed and 1.5 GB RAM. More time is spent in searching the neighborhood to improve the solution. The simulator takes less time in simulating the movement of trucks.

Construction heuristics randomly insert the customer sites in the trucks' routes to find an initial solution. The initial solution which might be feasible or non-feasible is slowly turned to a feasible solution using improvement heuristics by manipulating the customer sites in the routes of the trucks.

The developed system was tested against a small problem with optimum solution obtained using branch and bound integer programming algorithm. The performance parameter of total unloaded miles obtained by using simulated annealing approach was

found to be exactly the same as that obtained by integer programming approach even though the solutions were different.

The initial temperature for the annealing process is set at 100 degrees. The temperature above 100 degrees was found to have no impact on the solution except that it takes more time to find the solution. When the temperature was reduced below 100 degrees, there were more chances to restrict the solution to its local optima.

The number of local searches is fixed at 2000 at each temperature in our research. It should be increased when there are fewer trucks available for routing with high demand. The algorithm is allowed to proceed until the final temperature since the whole process takes less than 2 minutes.

There is an absolute necessity for timed deliveries of truck loads which are expected to increase in the future. Hence this research is very useful to meet the requirements of the forest supply chain.

Future research should be focused more towards generating a solution in limited and guaranteed time with fixed parameter setup and should be insensitive to changes in load requirements and the availability of trucks. The simulator can easily be made stochastic by generating and using random samples of the loading time, unloading time, traveling speed etc, which can produce a more practical solution. In the current research, the output from GPS system is entered into an Excel spreadsheet and then filtered data is fed to the program to generate a solution. Down the line, the program should be interfaced with the GPS system to directly process the data and deliver the solution. The

model then becomes sophisticated and requires advanced optimization techniques to find a solution.

The model should be made more dynamic accommodating future requests during plan execution. The tool would come in handy when there is an immediate additional requirement of loads to be delivered or when the truck serving the loads breaks down midway. In this case, the unavailable truck's responsibilities should be shared by other trucks. Sometimes due to weather uncertainties, the whole routing has to be altered.

The process of finding a minimum number of trucks required to meet the loads for the day is very complicated. One way of doing this is to reduce the number of vehicles to the point until the solution can be found within 10 minutes. If the tool takes more than 10 min to find the solution which is not desirable, the number of trucks is slightly increased by one and then solved to find the solution. The classification of a good logger or mill purely based on location can be done by allocating unit demand to all the loggers and mills.

The participating truck companies are required to make organizational changes in the method of operations to incorporate centralized decision making in a pooled log transport system. From an environmental point of view, fuel is saved since the truck travels fewer unloaded miles than needed and as a result more savings are realized in terms of fuel cost.

The demand matrix should be given to the centralized route scheduler early morning. The scheduler feeds the demand matrix to the program, and generates a best set of routes for the trucks. Truck drivers obtain the route and proceeds accordingly.

The system has been developed to increase the utilization of trucks, obey the shift duration of drivers, and to respect the time window at the logger and mill sites. The type of loading used here is hot loading since there is a loading and unloading time of 6 minutes.

The model can be used even when there is no hub by fixing the distance between the customer sites and the hub as zero. The logger sites, where the trucks first pick up the loads become the starting points and the mills, where the trucks delivers the load at the end of the shift becomes the ending point.

The program receives the demand requirements one hour prior to shift start time. The loads that are available only after few hours into the shift are most probably delivered by the trucks which would finish the given routing early or delivered the next day.

REFERENCES

- [1] Johanna C. Gerdessen 1996, "Theory and Methodology Vehicle routing problem with trailers", *European Journal of Operational Research* 93 (1996) 135-147
- [2] I-Ming Chao 2002, "A tabu search method for the truck and trailer routing problem", *Computers & Operations Research* 29 (2002) 33-51
- [3] Robert T. Sumichrast and Ina S. Markham 1995, "A heuristic and lower bound for a multi-depot routing problem", *Computers Ops Res.* Vol.22, No.10, pp.1047-1056, 1995
- [4] Sam R. Thangiah, Jean-Yves Potvin and Tong Sun 1996, "Heuristic approaches to vehicle routing with backhauls and time windows", *Computers Ops Res.* Vol.23, No.11, pp.1043-1057, 1996
- [5] Zbigniew J. Czech and Piotr Czarnas 2001, "Parallel simulated annealing for the vehicle routing problem with time windows", *Research grant BK-280-Rau2-2001*
- [6] Mikael Ronnqvist 2003, "Optimization in forestry", *Math. Program., Ser. B* 97:267-284 (2003)
- [7] Russell Bent and Pascal Van Hentenryck, "Dynamic vehicle routing with stochastic requests"
- [8] Andres Weintraub, Rafael Epstein, Ramiro Morales, Jorge Seron and Pier Traverso, 1996, "A truck scheduling system improves efficiency in the forest

- industries”, Institute for Operations Research and the Management Sciences, Interfaces 26: 4 July- August 1996 (pp.1-12)
- [9] M.Palmgren, M.Ronnqvist and P.Varbrand 2004, “A near-exact method for solving the log-truck scheduling problem”, International Transactions in Operations Research 11(2004) 447-464
- [10] Patrick Hirsch and Manfred Gronalt, “Tabu search based solution methods for scheduling log-trucks”
- [11] Russell Bent and Pascal Van Hentenryck, “A Two-stage hybrid local search for the vehicle routing problem with time windows”
- [12] Ibrahim Hassan Osman 1993, “Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem” Annals of Operations Research 41 (1993) 421-451
- [13] Jorge F.Valenzuela, H.Hakan Balci and Timothy McDonald 2003, “A Transportation-scheduling system for managing silvicultural projects” Journal of Forest Engineering 2003
- [14] Tim McDonald, Jorge Valenzuela “Optimizing Log Transport in the US South”

APPENDIX 1

LOGGER AND MILL NOTATION

Table A1.1-Notation of loggers.

| Notation | Name of the logger |
|----------|--------------------|
| 1 | Logger 1 |
| 2 | Logger 2 |
| 3 | Logger 3 |
| 4 | Logger 4 |
| 5 | Logger 5 |
| 6 | Logger 6 |
| 7 | Logger 7 |
| 8 | Logger 8 |
| 9 | Logger 9 |
| 10 | Logger 10 |
| 11 | Logger 11 |
| 12 | Logger 12 |
| 13 | Logger 13 |
| 14 | Logger 14 |
| 15 | Logger 15 |
| 16 | Logger 16 |
| 17 | Logger 17 |
| 18 | Logger 18 |
| 19 | Logger 19 |
| 20 | Logger 20 |
| 21 | Logger 21 |
| 22 | Logger 22 |

Table A1.2-Notation of mills.

| Notation | Name of the mill |
|----------|------------------|
| K | Mill 1 |
| M | Mill 2 |
| B | Mill 3 |
| C | Mill 4 |
| S | Mill 5 |
| T | Mill 6 |
| H | Mill 7 |
| O | Mill 8 |
| D | Mill 9 |
| G | Mill 10 |
| Q | Mill 11 |
| P | Mill 12 |
| F | Mill 13 |

APPENDIX 2

DISTANCE MATRIX BETWEEN LOGGERS AND MILLS

| Mill Log | K | M | B | C | S | T | H | O | D | G | Q | P | F | 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 |

APPENDIX 3

DEMAND MATRIX BETWEEN LOGGERS AND MILLS

| Log \ Mill | K | M | B | C | S | T | H | O | D | G | Q | P | F |
|------------|---|---|---|---|---|---|----|---|---|---|---|---|---|
| 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 |