Final Report

Combined Truck Routing and Driver Scheduling Problems under Hours of Service Regulations

by

Hokey Min*

Affiliation: James R. Good Chair in Global Supply Chain Strategy
Department of Management, BAA 3008C
College of Business Administration
Bowling Green State University
Bowling Green, OH 43404
Telephone: 419-372-3442 (office)
419-372-6057 (fax)
E-mail: hmin@bgsu.edu

Prepared on August 20, 2009

* Principal Investigator

Executive Summary

Regardless of changing variants, hours-of-service (HOS) regulations are intended to help truck drivers ensure get adequate rest and perform safe operations. The new HOS regulations, however, may lead to substantial cost increases for regional common carriers which have already been hit hard by rising fuel prices and declining shipping demands. In addition, the new HOS regulations complicate driver schedules by not only restricting the driver's consecutive driving hours, but also expanding off-duty hours. To deal with this complex challenge, we develop a mixed-integer programming model and a simulated annealing (SA) meta-heuristic for solving that model. To validate the practicality and efficiency of the proposed model and heuristic solution procedure, they were applied to actual truck routing and driver scheduling problems encountering a regional common carrier. A series of computational experiments and sensitivity analysis with actual truck routing and driver scheduling problems verified the solution accuracy and computational efficiency of the SA meta-heuristic.

1 BACKGROUND

The leading cause of truck accidents is driver fatigue. In fact, driver fatigue was the primary cause of 2% to 23% of all truck crashes (O'Hanlon 1978; Horne and Reyner, 1995). For example, Reissmann (1997) discovered that drowsy drivers were responsible for 50% of the fatal vehicle crashes on the Pennsylvania Turnpike and New York Thruway. This alarming statistics is not surprising given that drivers who have been awake for 24 hours have an equivalent driving performance to a person who has a BAC (blood alcohol content) of 0.1 g/100ml and thus are seven times more likely to have an accident (http://www.smartmotorist.com/traffic-and-safety-guideline/driver-fatigue-is-an-important-cause-of-road-crashes.html, 2008). Recognizing the seriousness of driver fatigue to highway safety, Federal Motor Carrier Safety Administration (FMCSA) has attempted to implement new hours-of-service (HOS) regulations. The new HOS regulations effective on October 1st of 2005, however, may lead to substantial cost increases for the trucking industry which will in turn hurt shippers and ultimately customers. For instance, the trucking industry may need to hire additional 84,000 drivers to comply with the new HOS rules requiring that drivers be placed out-of-service until they accumulated enough off-duty time. To elaborate, off-duty breaks required to refresh driving hours were increased to 10 consecutive hours from the old rule of eight cumulative hours. A chronic shortage of truck drivers coupled with new HOS regulations could further complicate the driver scheduling problems. In addition, due to potential loading/unloading delays and stiffer fines/penalties (between $550 and $11,000 per violation depending on the severity) resulting from new HOS regulations, trucking firms will be faced with daunting challenges of controlling mounting costs while complying with new HOS regulations.

In an effort to help trucking firms cope with these challenges, this report aims to develop a mathematical model and its solution procedure that can minimize transportation costs and avoid the driver schedule conflicts, while meeting new HOS regulatory requirements and satisfying truck capacity constraints. The proposed model takes the form of the combined time dependent vehicle routing and driver scheduling problem that is concerned with finding an optimal route/schedule of capacitated vehicles and an optimal assignment of drivers to such vehicles over a number of designated delivery points within pre-specified time windows. The combined time dependent vehicle routing and driver scheduling problem may arise in many real-world situations such as just-in-time (JIT) delivery services, overnight trucking services, express parcel delivery services, and school bus services. Similar to other well-known vehicle routing and scheduling problems, the combined time dependent vehicle routing and driver scheduling problem is extremely difficult to solve due to its combinatorial nature and added complications, such as time windows and simultaneous scheduling of both drivers and their vehicles. As such, the combined time dependent vehicle routing and driver scheduling problem calls for heuristic solution procedures that can handle practical-size problems faced by many trucking firms. In this report, we propose simulated annealing (SA) meta-heuristics as a way to solve the combined time dependent vehicle routing and driver scheduling problem.

2 PRIOR LITERATURE

Although there exists abundant literature dealing with various forms of vehicle routing and scheduling problems, a relatively few attempts have been made to solve time dependent vehicle routing and scheduling problems (TDVRSP) that is concerned with the determination of optimal routes by considering the time it takes to traverse each given arc depending on the time of the day. Some notable examples of these attempts include: Malandraki and Daskin (1992) and Donati et al. (2006). To elaborate, Malandraki and Daskin (1992) proposed a mixed integer programming model and cutting plan heuristics that took into account time windows and the maximum allowable duration for each route (e.g., work schedules of the driver). The travel distance of arc (i, j) is a time dependent step function as the speed of the vehicle does not remain constant due to the variable traffic density. TDVRSP provided substantial improvements over a vehicle routing and scheduling problem based on fixed travel times. Donati et al. (2006) extended the earlier work of Malandraki and Daskin (1992) by not only considering the vehicle speed, but also developing ant colony optimization meta-heuristics to improve the computational efficiency and accuracy of the solution procedure. Similar to most of the other studies dealing with TDVRSP, these two studies, however, did not take into account simultaneous vehicle routing and driver scheduling problems. For the further details of TDVRSPs that were solved in the past, the interested readers should refer to Bodin et al. (1983) and Solomon and Desrosiers (1988).

On the other hand, Portugal et al. (2006) focused on the driver scheduling problem under strict labor rules. They solved the driver scheduling problem by using a set partitioning model that consisted of two phases: (1) generation phase; (2) resolution phase. The generation phase developed a feasible set of driver duties for based on the parameters defined. Labor rules, security procedures, and planning strategies can be used to generate a set of feasible duties during the generation phase. In the resolution phase, a subset of feasible duties is selected on a particular schedule to minimize cost. However, they did not integrate the driver scheduling problem with the vehicle routing/scheduling problem and thus overlooked the potential schedule conflict between the driver and the vehicle assigned to that driver. Xu et al. (2003) were among the first to consider old HOS regulations for a multiple vehicle routing and scheduling problem with time windows. In addition, they imposed a set of compatibility constraints that specified which orders could not be covered by which carrier/vehicle types and which orders could not be shipped together. Order loading and unloading sequence must satisfy the nested precedence constraint that required an order to not be unloaded until all the orders are loaded into the truck. Each vehicle trip must satisfy the driver's work rules prescribed by the U.S. Department of Transportation which specified legal working hours of a driver. The cost of a trip is determined by several factors including a fixed charge, total mileage, total waiting time, and total layover time of the driver. To solve this complicated problem, they formulated a set partitioning model and developed a column generation procedure. However, their formulation did not explicitly take into account HOS regulations and a varying speed of the vehicle. Similar to Xu et al. (2003), Goel and Gruhn (2006) considered drivers' labor rules in the European Union and embedded such rules into driver scheduling problems. They also formulated a vehicle routing problem with time windows to incorporate driver scheduling issues into the vehicle routing/scheduling problem. Their study, however, did not explicitly consider the vehicle speed for the total travel time of the vehicle. More recently, Archetti and Savelsbergh (2007) took into account HOS rules in the trip scheduling problem. Given a sequence of n transportation requests with dispatch windows at the origins, they determined a driver's schedule (if it exists), i.e., driving times and rest times, so that the origins could be visited in the given sequence and within their dispatch windows, or otherwise it was found that such a feasible schedule did not exist. They assumed that loading and unloading at customer locations are performed instantaneously and waiting time at the locations could be converted into daily break time. They develop a backward search algorithm with $O(n^3)$ polynomial time to solve the trip scheduling problem. However, their study neglected time-dependent travel times, since they presumed that vehicle travel time was fixed regardless of local traffic congestions and speed limits. Another shortcoming is that their model and algorithm failed to consider complex HOS regulations regarding the scheduling of restorative breaks during the day, although they could handle HOS regulations concerning night time rest.

To go beyond these prior studies, we attempted to solve a multiple vehicle TDVRSP under the most recent HOS rules. The main features of the proposed model and solution procedure are:

- Until recently, most of the existing vehicle routing and scheduling literature focused solely on the minimization of travel distances, travel time or transportation cost under the premise of constant vehicle speed. Similar to Malandraki and Daskin (1992) and Donati et al. (2006), we developed a step function with consecutive time intervals that took into account changes in vehicle speed due to traffic congestions and road accidents. In other words, the proposed model and solution procedure can factor variable vehicle speed into driving time and thus prevent sub-optimal or infeasible vehicle and driver schedules.
- Considering that driver's working hours may not match the available vehicle schedules, we attempted to simultaneously coordinate both driver and vehicle schedules. These attempts are very rare in the literature due to inherent computational complexity involved in simultaneous driver and vehicle scheduling.
- As in the earlier works of Xu et al. (2003), Goel and Gruhn (2006), and Archetti and Salvesbergh (2007), we incorporated HOS regulations into the proposed model and solution procedure. Especially, unlike those earlier attempts, the proposed model and solution procedure explicitly considered the most recent HOS regulations enacted on October 1st of 2005.

As specified above, the proposed model and solution procedure are capable of capturing the aforementioned realistic dimensions, while other existing models could not.

3 PROBLEM STATEMENT

Consider a truck (tractor-trailer) with a full truckload of goods that should be delivered to a number of customer locations across the United States. A majority of these customer locations are more than 500 miles away from the truck's home depot (domicile). Thus, long-haul drivers are required to travel the average of 500-600 miles a day. Such a driving requirement forces the drivers to spend most of their time (including breaks, rests, and sleeps) on the road. Typically, the long-haul drivers are returning home weekly, staying at home for one night per week. A stretch of cumulative long working hours on the road makes long-haul driving extremely stressful. In addition to job stress, long-haul driving significantly increases the chance of truck crashes. In the long-haul sector, truck driver fatigue is 18 times greater than that of the short-haul sector (see, e.g., Lowe, 2007). Since driver fatigue is one of the leading causes of truck crashes, Federal Motor Carrier Safety Administration (FMCSA) stipulated a series of hours of service regulations that restrict consecutive hours of driving and mandate minimum hours of restorative breaks and sleeps. For example, revised HOS regulations of 2005 require the driver to:

- Drive a maximum of 11 hours after 10 consecutive hours off duty;
- Not exceed the 14 hours of driving after coming on duty, following 10 consecutive hours off duty;
- Not drive after 60/70 hours on duty in 7/8 consecutive days. A driver may restart a 7/8 consecutive day period after taking 34 or more consecutive hours off duty;
- Take 10 hours off-duty at the sleeper-berth, but may split sleeper-berth time into two periods provided neither is less than 2 hours.

The complexity of these regulations coupled with chronic driver shortages creates a scheduling nightmare for truck dispatchers who are responsible for scheduling the driver's working hours in such a way that he/she can spend more time at home, while meeting their customers' delivery deadlines. Since driver schedules cannot be completed without assigning each driver to an available truck, both driver and truck schedules should be coordinated together and developed simultaneously. In other words, a driver is assigned to the same truck during his entire duty hours and his/her schedules are tightly dependent on predetermined truck schedules and routes or vice versa. In a nutshell, the combined truck routing and driver scheduling problem under HOS regulations (CTRDSP-HOS) is primarily concerned with the minimization of the total working hours of drivers and travel time of trucks given a set of routes within a fixed time horizon. Each route has fixed starting and ending times and is assigned to a truck and a driver from a certain set of domiciles. Each route contains a number of delivery nodes (i.e., customer locations) where partial shipments will be dropped off within specific time windows set by customer delivery preferences or requirements. Herein, notice that the arrival time at a customer location depends on the departure time from the preceding customer location. It is also affected by a myriad of factors: (1) local traffic congestions and speed limits on the roadway network between those two customer locations; (2) required restorative breaks during the trip between those two locations; (3) unexpected unloading delays at the preceding customer location. Furthermore, in an effort to increase more time at home for drivers, remote domiciles can be used to reposition the drivers to keep them near their domiciles. These unforeseen circumstances and realities of HOS regulations may necessitate a modification of truck routes in such a way that driver waiting or truck idle time can be minimized while early and/or late arrivals should be avoided.

In particular, to reflect a variable vehicle speed during a different time of the day, we develop a step function that can create different speed distribution with 24 time intervals where each time interval corresponds to a particular hour of the day. The step function can be mathematically expressed as: $c(t) = c_{ij}^{h}$, $h \le t \le h+1$, where $c(t)$ is a speed distribution at the time (t) of the day and c_{ij}^{h} is the speed of the vehicle during the h^{th} hour (i.e, $h = 0, 1, 2, 3\ldots, 23$) of the day for a particular customer link (i, j).

As described above, a determination of what sequence a vehicle should traverse is complicated by customer requests for timely delivery, varying travel speed, and compliance with HOS regulations. Thus, the CTRDSP-HOS is considered a special case of the time-dependent, multiple vehicle routing/scheduling problem with an added complexity of a driver scheduling problem subject to HOS regulations.

4 MODEL DESIGN

The truck starts its tour at the depot, visits all customers, and returns to the depot. Each customer is associated with a node on the transportation network. Although it may be the same geographical location, the depot is associated with two nodes, the origin depot ($k = 0$) and the destination depot ($k = n+1$). The objective function here is to minimize the total time to complete a tour. Starting at node 0, visiting n customer nodes and terminating at node n+1. The total time is the sum of travel time, waiting time and service time at the customer node, and needed break time during the trip.

4.1 Model Assumptions

Prior to developing a mathematical model, we made the following assumptions:

1. The driver can be assigned to only one truck. Also, relay driving is not considered.
2. Service (unloading) time at each customer node is known a priori.

3. Once the driver starts unloading services at a customer location, he/she is not allowed to take a daily restorative break before the completion of unloading services.
4. Daily restorative breaks can be taken at any locations during the designated trip.
5. Travel distances/times between two nodes are Non-Euclidean and asymmetric.

4.2 Model Formulation

4.2.1 Indices and sets

G (V, E) = Graph representing the transportation network with a set of nodes V and set of edges E

A= set of customer nodes, A= {1, ….., n}, $A \subset V$

P_{kl} = set of paths from node $k \in A$ to node $l \in A$, $k \neq l$ on G (V, E)

P_{kl}^r = the r^th path from customer node k to customer node l, $P_{kl}^r \in P_{kl}$

P_{kl}^* = shortest time path from customer node k to customer node l

4.2.2 Model Parameters

S_i = state of the driver at node i (i.e., accumulated driving time after the most recent break, accumulated duty time after the most recent break, accumulated weekly duty time after the most recent weekly break)

$t_{ij}(t_i, S_i)$ = travel time on arc (i, j), when vehicle leaves node i at time t_i with driver at state S_i

$t_{P_{kl}^*}(t_k, S_k)$ = shortest time to reach customer node l from customer node k, when vehicle leaves node k at time t_k with the driver at state S_k

s_l = service time at customer node l (s_l =0, for l = 0 and n+1)

t_0 = the starting time from the origin depot (node 1)

M = arbitrarily large number

a_l = earliest time that the vehicle can start service at node l

b_l = latest time that the vehicle can start service at node l

4.2.3 Decision Variables

$$y_{kl} = \begin{cases} 1, \text{if vehicle visits customer } l \text{ immediately after customer } k \\ 0, \text{otherwise} \end{cases}$$

$$x_{ij} = \begin{cases} 1, \text{if vehicle traverses arc } (i, j) \text{ of the transportation network} \\ 0, \text{otherwise} \end{cases}$$

t_j = departure time of vehicle from node $j \in V$. The starting time of the trip from node 0, t_0 is given, as well as the state of the driver, $S_1 = \{0, 0, 0\}$

4.2.4 Mathematical Formulation

Minimize $t_{n+1} - t_0$ (1)

Subject to

$$t_{P_{kl}^*}(t_k, S_k) = \min_{P_{kl}^r \in P_{kl}} \sum_{(i,j)\in P_{kl}^r} t_{ij}(t_i, S_i) x_{ij} \qquad k \in A \cup \{0\},\ l \in A \cup \{n+1\},\ k \neq l \tag{2}$$

$$t_l - t_k - M y_{kl} \geq t_{p_{kl}^*}(t_k, S_k) + s_l - M \qquad k \in A \cup \{0\},\ l \in A \cup \{n+1\},\ k \neq l \tag{3}$$

$$a_l \leq t_l - s_l \leq b_l \qquad l \in A \tag{4}$$

$$\sum_{j \in V} x_{kj} = 1 \qquad k \in A \cup \{0\} \tag{5}$$

$$\sum_{j \in V} x_{jl} = 1 \qquad l \in A \cup \{n+1\} \tag{6}$$

$$\sum_{i \in V, i \neq j} x_{ij} - \sum_{k \in V, k \neq j} x_{jk} = 0 \qquad j \in V \tag{7}$$

$t_j \geq 0 \qquad j \in A \qquad (8)$

$x_{ij} \in \{0,1\} \qquad (i, j) \in E \qquad (9)$

$y_{kl} \in \{0,1\} \qquad k \in A \cup \{0\},\ l \in A \cup \{n+1\}, k \neq l \qquad (10)$

The objective function (1) is to minimize total tour time i.e., the sum of travel time, traffic delays, break time, waiting time and service time to complete a tour. Given the departure time from node *k* and the state of the driver (t_k, S_k), constraint (2) states that if customer *l* is visited after *k*, then the shortest path from node *k* to node *l* is selected. The total time to traverse that path is the sum of travel time along the shortest time path from customer node *k* to customer node *l*, including the break time if necessary. Although it is not explicitly specified in the formulation, driver's break time is included, if necessary. Constraint (3) determines the departure time from customer node *l* if visited after customer node *k* ($y_{kl} = 1$). It is the sum of arrival time at customer node *l* plus the service time at node *l* plus the slack which is break time/waiting time. Note that when y_{kl}=0, this constraint is always satisfied. Constraint (4) ensures that the start of the service at customer node *l* should be within the time window. If the driver arrives at a customer location before the start of the time window, he/she must wait until the beginning of the time window. Constraints (5) and (6) ensure that there is only one outgoing arc from customer node *k* and only one incoming arc into customer *l* respectively, or equivalently, a feasible tour should include all customer nodes. Constraint (7) describes the balance equations for all nodes *j* of the transportation network. Constraint (8) states that departure time from customer node *j* should be non negative. Constraints (9) and (10) designate both x_{ij} and y_{kl} as binary variables.

5 MODEL TESTING AND RESULTS

To solve the proposed model described in the previous chapter, we developed two-phase solution procedures comprising a Time Dependent Dijkstra's (TDD) algorithm and a simulated annealing meta-heuristic as shown in Figure 1. To elaborate, the initialization phase begins with a greedy heuristics which was designed to develop a Hamiltonian tour for visiting all customers and then returning to the depot. In each iteration, the greedy heuristic finds the shortest time path to visit each customer l from the current customer (depot) k using the time dependent Dijkstra's algorithm. Since the travel time along each arc is not fixed, a time dependent travel function with HOS is incorporated into Dijkstra's algorithm to find the travel time along each arc (i, j). After the arrival time at each customer node l is found, the departure time is computed by adding service time, waiting time if arrival occurs outside the service time window of the customer, and break time if necessary. The next customer location (after k) to be visited is selected among the customers who have not been visited yet using the customer selection function. The above process is repeated until we find an initial tour. In the improvement phase, we use a simulated annealing schedule to improve the solution iteratively until a certain halting criterion is met. The improvement phase generates a near optimal solution.

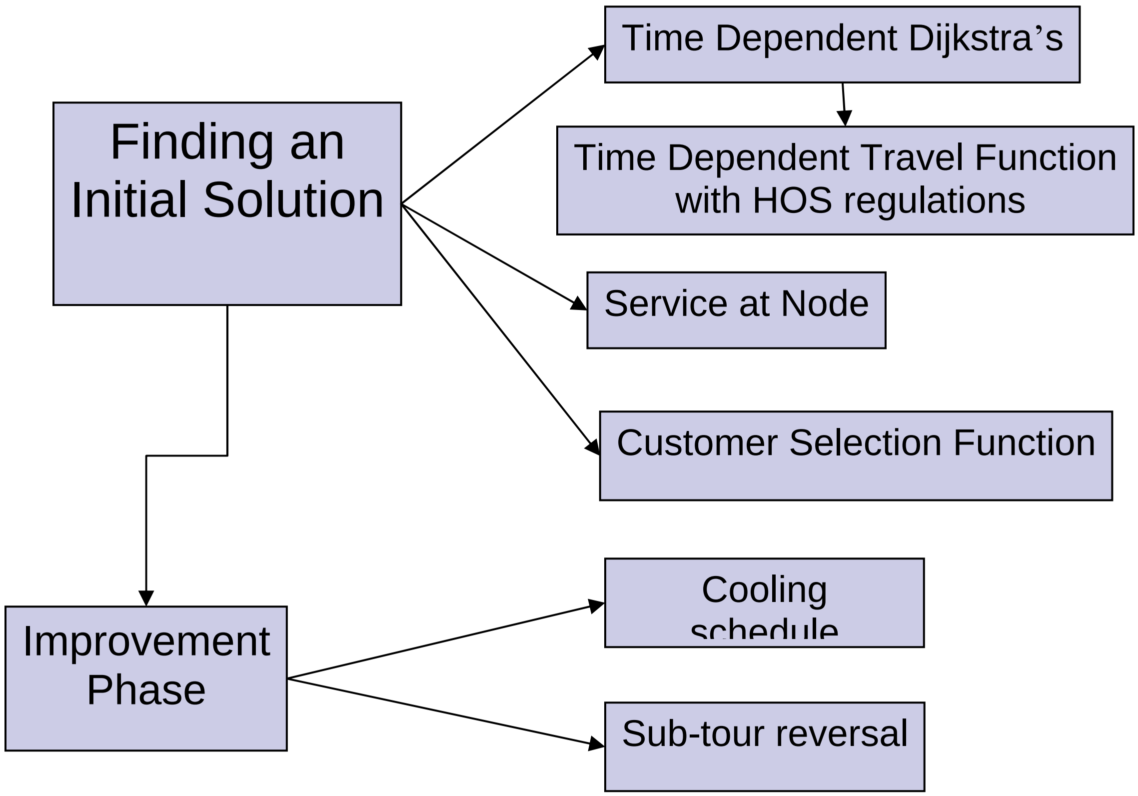

Figure 1. A Structure of the Proposed Solution Procedures

5.1 Time Dependent Dijkstra's Algorithm (TDD)

Time Dependent Dijkstra's algorithm (TDD) is implemented to calculate the shortest time path P^*_{kl} between two customer nodes *k* and *l* (i.e., node *k* is the origin and node *l* is the destination) on a transportation network. In this paper, we have modified the Dijkstra's algorithm by incorporating time dependent travel with HOS regulations.

The trip is initiated at the origin depot. The starting time of the trip is set to time t_0. The set of customers/depot nodes are connected through the highway arcs (set *E*) of the transportation network. Actually there are many paths linking any two nodes $k, l \in A$ (set P_{kl}). In finding the shortest time path P_{kl}^*, the TDD employs the "Time Dependent Travel function under HOS regulations" that took into varying time to traverse an arc (i, j). This is because the vehicle speed may be changing during the arc traversal or the driver may need a break. In the ordinary Dijkstra's algorithm, the time to traverse an arc (i, j) is considered constant. Given the departure time from node i, t_i and the state of the driver $S_i = \{drt, dut, wdut\}$, the function to calculate the arrival time at node j is shown in Figure 2 and the notation used is explained below. Assuming that the length of the trip does not exceed a week, we can simplify the notation by truncating the third component of the state of the driver, weekly duty time. The functions and the algorithm can be easily extended to the case where a trip may take more than week.

The notations for time dependent travel function with HOS regulations on arc (i, j) is as follows:

t: current time at node *i*

drt: accumulated driving time since last daily break

dut: accumulated duty time (driving time, service time and waiting time) since last daily break

wdut: accumulated weekly duty time since last weekly rest period

daily_max: maximum daily driving time between consecutive daily break periods (11 hours)

duty_max: maximum duty time between two consecutive duty break periods (14 hours)

weekly_max: maximum weekly duty time between two consecutive weekly rest periods (36 hours)

daily_break: time required for a daily break period (10 hours)

res_ddr: residual driving time before the next daily break period (i.e. *daily_max* – *drt*)

res_dut: residual duty time before the next daily break period (i.e. duty_*max* – *dut*)

res_wdr: residual weekly driving time before the weekly break period (i.e. *weekly_max* –

wdut)

In Figure 2, the vehicle leaves node *i* at time *t*. The travel speed of the vehicle depends on the time of the day it departs from node i (i.e. v (t) = $c_{ij}^{h} \mid h \leq t \leq h+1$). The arrival time at node *j* depends on the speed of departure from node *i*. The state of the driver at node *i*, S_i is defined as a vector with two components: (1) the accumulated driving time; (2) the accumulated duty time. The residual daily driving and, daily duty are computed next. Then, the earliest time δt is computed until one of the following events occurs: (a) speed on arc (*i*, *j*) changes, (b) node *j* is reached, (c) maximum daily driving time is reached, and (d) maximum duty time is reached. Whichever event occurs first, the current time and the state of the driver are updated. The recursive function is executed repetitively and the state of the driver is updated.

If the accumulated driving time or duty time is not sufficient to reach node *j*, the driver takes a daily break. Time of departure after taking the break is updated and daily driving (*drt*) and duty time (*dut*) are reset to zero. The speed to traverse the remaining distance depends on the time of the day the driver departs after taking the break. The above mentioned steps are repeated until node *j* is reached. The shortest path P_{kl}^{*} from customer node $k \in A$ to customer node $l \in A\text{-}P$ is calculated using TDD.

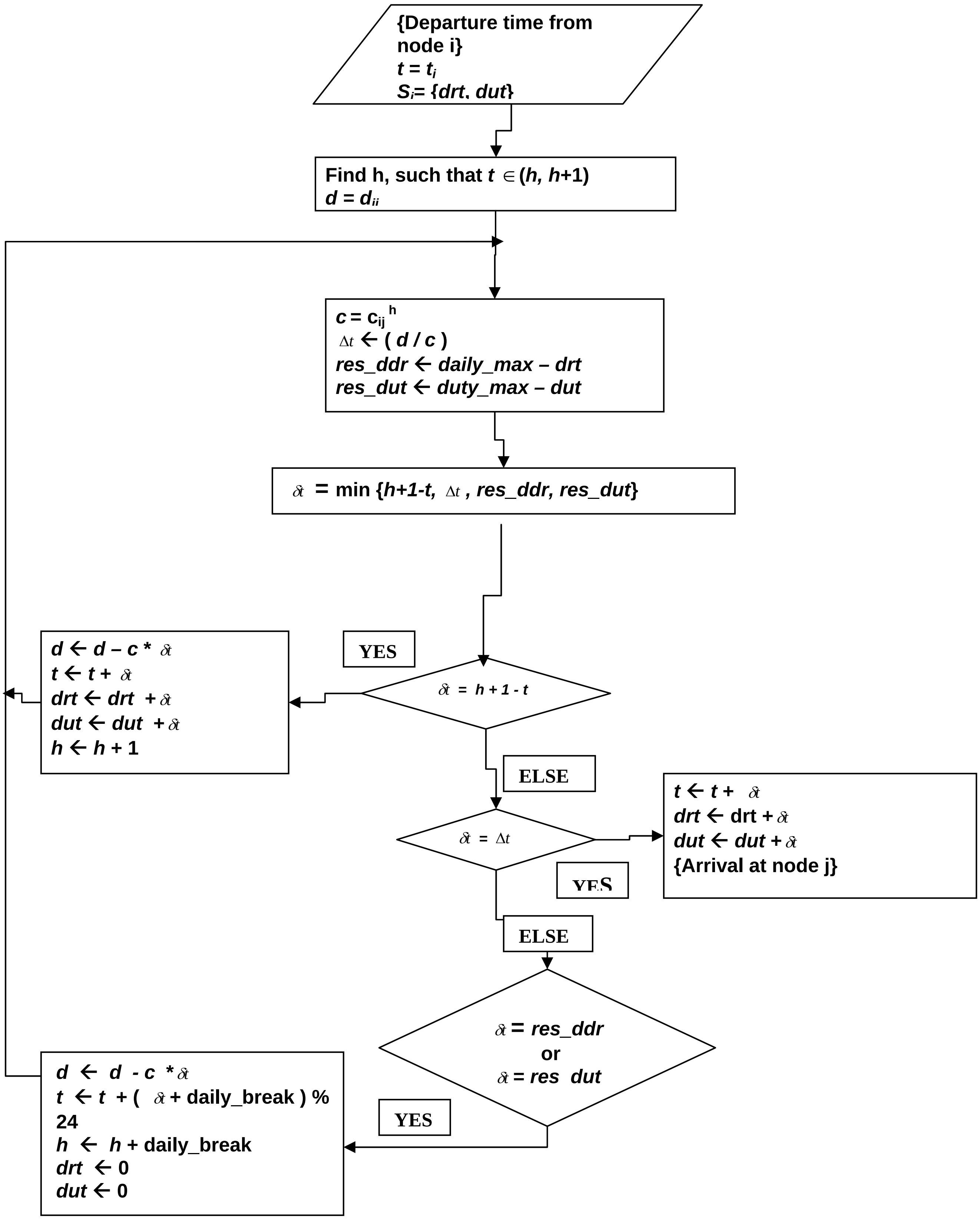
{Departure time from node i}
t = t_i
S_i = {drt, dut}
Find h, such that t ∈ (h, h+1)
d = d_ij
c = c_ij^h
Δt ← (d / c)
res_ddr ← daily_max – drt
res_dut ← duty_max – dut
δt = min {h+1-t, Δt , res_ddr, res_dut}
δt = h + 1 - t
YES
d ← d – c * δt
t ← t + δt
drt ← drt + δt
dut ← dut + δt
h ← h + 1
ELSE
δt = Δt
YES
t ← t + δt
drt ← drt + δt
dut ← dut + δt
{Arrival at node j}
ELSE
δt = res_ddr
or
δt = res dut
YES
d ← d - c * δt
t ← t + (δt + daily_break) %
24
h ← h + daily_break
drt ← 0
dut ← 0

Figure 2. Time dependent travel function with HOS regulations

5.2 Simulated Annealing Heuristic

Simulated annealing generates solutions in the neighborhood of the current solution and evaluates them. A random generation method is used to select two nodes from the current solution. The two nodes are swapped to obtain a new solution. According to Connolly (1990), this approach might be inefficient as potential improvements might be missed at lower temperatures due to the random nature of node selection. Thus, the solution quality depends on node selection. However, various schemes can be used to improve efficiency of simulated annealing as described below:

5.2.1 Initial temperature

Following the suggestion made by Connolly (1990), we performed M = 50 k [where k = 0.5 (n) (n-1)] random swaps initially to find δ_{max} (Largest uphill step) and δ_{min} (smallest uphill step). The initial temperature T_0 is set to δ_{max} (thereby giving an initial probability greater than 0.4). It is calculated using the following formula:

p = e^ [(evaluation (current solution) - evaluation (new solution))/T_0

5.2.2 Cooling schedule

The cooling schedule controls the rate at which the temperature changes and helps the heuristic to avoid local minima. It causes the heuristic to act more erratically when the temperature is high and consequently, at higher temperatures, the probability of accepting the worse solution is much higher than that at a lower temperature. The cooling schedule can be linear, exponential or polynomial. The cooling schedule can be determined by trial and error based on the trade-off between the solution quality and the computational time. If we decrease the temperature too quickly, the system will get "quenched." That is to say, the system is still in the higher energy state (higher objective function value) and the temperature is too low to find a tour with lower energy state (lower objective function value). Thus, we may end up with local minima and the algorithm may not be able to find a good solution. At each temperature t_k, after reaching the equilibrium, we reduce the temperature by γ. Thus $t_{k+1} = \gamma * t_k$. The range of γ is between [0.5, 0.9]. The different cooling schedules were used and then evaluated with respect to solution quality. To find near-optimal solutions, we used $\gamma = 0.9$ in the computational experiments.

5.2.3 Transition mechanism

The *n* customers can be arranged in *n!* permutations. Notice that a reverse order in a permutation yields a different solution, because travel distances (times) between two nodes are not necessarily symmetric due to varying travel times. It is possible to generate all permutations of given *n-1* customer nodes within finite time. However, the computational time required by the enumeration method is increasing exponentially with *n*. A sub-tour reversal (i.e. exchanging of two nodes in a given tour) generates new tours. The new tour generated is evaluated and total time to complete the tour is compared with the previous tour. The sub-tour reversal requires a selection of both the beginning and ending slots from a given sequence of customer visits. The beginning slot and ending slot can be anywhere except from first and the last slot from the given sequence. Random numbers are generated to select the beginning and ending slot. The departure time in the new tour from all nodes preceding the beginning slot remains unchanged. The departure times from the remaining nodes have to be recalculated using the TDD algorithm.

5.2.4 Equilibrium condition required for reaching a steady state at each temperature

A number of iterations were made at each temperature to reach an equilibrium condition. In our experiments, we used Burkard and Rendl's formula (1984) that requires L = 0.5 * n^2 (where$n$ is the number of customers) iterations at each stage to reach an equilibrium. Also, we multiplied L by 1.1 to improve solution quality. Through a series of experiments, we found that a multiplicative factor of 1.03 produced results as good as those obtained by using a multiplicative factor of 1.1. The decrease in a multiplicative factor tended to reduce computational time considerably.

5.2.5 Stopping rule

Once the halting criterion (minimum temperature) is reached, the SA heuristic will stop. The halting criterion is denoted as $\delta_{min.}$ The heuristic can be stopped earlier after certain number of iterations (approximately after 3200 iterations), if the cumulative gap between the current solution and the best solution obtained so far for the last 5 consecutive iterations is less than or equal to a minimum reduction rate (ε). In our experiments, either a minimum reduction rate or a halting criterion, whichever occurs first, was used as the stopping threshold.

Pseudocode for SA (for minimization)

Procedure simulated annealing

Begin

Current solution

best solution ← current solution

initialize temperature *T*

initialize halting criteria

while (temperature > halting criteria)

for i:= 1 to50 do

```
        select randomly two nodes from current solution and swap them
        Evaluate new solution
        if evaluation(new solution) < evaluation(current solution)
            current solution ← new solution
        else if random[0,1) < e^((evaluation(current solution)-evaluation(new solution))/T)
            current solution ← new solution
        if evaluation(current solution) < evaluation(best solution)
            best solution ← current solution
        end for loop
    temperature ← schedule(temperature)
    i = integer part of (50*1.03)
    if ( cumulative difference of the last four stages < 0.01)
        break
end
```

Simulated annealing starts with an initial solution. At each stage k, a number of iterations are made to attain equilibrium. The new configuration is generated by randomly selecting two nodes and swapping them. A change in the total travel time is computed. The evaluation of the trip is done starting from the customer where the change has taken place. The new solution is always accepted if the objective value E_1 is less than the objective function value of the current solution E_2. The current configuration then becomes the new configuration. If the objective function value $E_1 > E_2$, the new solution is accepted based on the probability, p = $\exp\left(\frac{E_2 - E_1}{T}\right)$. The new solution is then compared with a random number generated with a uniform probability distribution in [0, 1). If r < p = exp $\left(\frac{E_2 - E_1}{T}\right)$, then the new solution is accepted. The current solution is updated to the new solution. If the random number generated is greater than the given probability, we do not have to update the current solution. Once the equilibrium is reached, temperature is lowered by γ to T.

5.3 Test Results

To check the efficiency of the proposed solution procedures, we compared the optimal solutions obtained by exhaustive enumeration to the solutions obtained from the SA heuristic for real-world problems encountered by a regional truckload (TL) carrier which primarily serves customers in the states east of Mississippi. To keep the confidentiality of this carrier, it will be called "Delta." Delta has a fleet of refrigerated, temperature-controlled trucks which haul consumer retail products including foodstuff commodities (e.g., frozen ice cream, fresh meats, various produce). To illustrate the Delta's current operations, the test problems were generated based on key customer bases in Northeastern United States. These customer bases are summarized in Table 1.

These customers are served by the central warehouse (depot) located in Wilbraham, MA, (highlighted in Table 1). From this depot, the truck departs at 7:00 am on Monday and returns back after serving customers located in 14 different towns. The domicile of the driver is at Enfield, MA, which is visited during the tour and the driver stays for a period of at least four hours starting at some time during the time window 2pm – 10 pm. The time window for starting service at the remaining thirteen towns (customers) is 9 am – 5 pm. Service time at each customer is fixed to two hours.

City or Town	Abbreviation
1. Assonet, MA	Asso
2. Brattleboro, VT	Brat
3. Cheshire, CT	Che
4. Ellington, CT	Ell
5. Enfield, CT	Enf
6. Hartford, CT	Hart
7. Long Meadow, MA	LM
8. Methuen, MA	Meth
9. New Britain, CT	NB
10. New Haven, CT	NH
11. Revere, MA	Rev
12. Sturbridge, MA	Stur
13. Westfield, MA	West
14. Wilbraham, MA	**Wil**
15. Worcester, MA	Wor

Table 1. Customer locations

A map of the Northeast United States that displays the underlying transportation network is shown in Figure 3. Only highway segments that may be traversed during the tour have been considered. The resulting network consists of 52 nodes and 138 arcs. Of the 52 nodes, 15 represent the cities or towns listed in Table 1 and the remaining nodes represent highway intersections. The arcs represent highway segments between nodes. The arcs with their distances are listed in Appendix A. The arcs are directed, i.e., each highway segment is represented by two arcs. For example, the first two lines in Appendix A refer to the highway segment of Interstate 495 (I495) between the town of Methuen and the intersection of I495 and I93, having length 16 miles.

Figure 3. Underlying transportation network connecting customers and depot

Highway traveling speed data for each hour of the day were estimated from Google Maps (*http://maps.google.com/*). Their *traffic* feature displays for each hour of the day the highway speed with different colors, corresponding to different speed levels from slow to fast. Three different speed levels were estimated at 20, 40 and 60 miles per hour. When parts of a highway segment (arc) were colored differently in a given hour of the day, the different speeds were weighted by the respective lengths of the highway subsegments for an overall effective speed of the arc. The resulting time dependent speed data for each arc of the network and each hour of the day starting at 12 o' clock midnight are displayed in Appendix B. For example, the speed between 12:00 midnight and 1:00 a.m. on Highway *MethuenI495_I93_via_I495* (first line) is 65 miles per hour (first entry) while between 7:00 and 8:00 am is 58 miles per hour (eight entry).

The above data were used as input into the *Time Dependent Truck Routing and Driver Scheduling Model* and the *Simulated Annealing* (SA) *Metaheuristic*, described in a previous report. Table 2 summarizes the results. Different initial solutions (different starting points in the solution space) were obtained using different values of α, $\alpha = 0.3$, 0.5, 0.7 and 0.9. For each initial solution, simulation annealing was performed using four different schedules with $\gamma = 0.7$, 0.8, 0.85 and 0.9, for a total of 16 experiments for all the combinations of values of α and γ. Of the generated 16 routes, 14 are distinct, ranging from the best objective value of 88.4 (highlighted in Table 2) to the worst value of 92.11, i.e. within a 3.71 hours time interval. Each SA run took on the average 12-13 minutes for $\gamma = 0.85$ on a Dell Intel core 2 Duo, 2.16 GHz, 4 GB RAM computer. Finding the exact optimal solution by evaluating all $(n-1)! = 14! = 8.72\text{x}10^{10}$ possible permutations of the towns was attempted, however, it was unsuccessful. The computer was running for more than a week without completing the enumeration. Nonetheless, based on the extensive computational experiments and sensitivity analysis we conducted in testing the metaheuristic (please refer to section 5.4), we feel confident that the best solution obtained by the SA procedure, if not optimal, it is very close to the optimal.

The best route found (abbreviated in Table 2) is: Wilbraham - Sturbridge - Worcester - Brattleboro - Enfield - Cheshire – New Haven – New Britain - Hartford - Revere - Methuen -

Assonet - Westfield – Long Meadow - Ellington – Wilbraham. The truck route and driver's schedule are summarized in Appendix C. Note that for notational simplicity and computational efficiency, fractional times are used with rollover from one day to the next (cumulative time). The truck leaves the depot at Wilbraham at time 7.00 and returns back at time 88.40. This translates to departure time from Wilbraham at 7:00 am on Monday and arrival back at 4:40 pm on Thursday for a total tour time of 81 hours and 24 minutes. As it is shown in Appendix C, the driver takes off-duty break on each one of the three nights, on Monday night at his home in Enfield, CT, on Tuesday night at Revere, MA, and on Wednesday night at Westfield, MA. The truck waits after the off-duty break and before service starts at 9:00 am, in Sturbridge on Monday morning, in Revere on Wednesday morning and in Westfield on Thursday morning. The driver's schedule looks very reasonable, having all off-duty breaks overnight including one at home.

Weight (α)	Initial objective value	Cooling Schedule (γ)	Objective value of SA metaheuristic	Route
0.3				
	124.8	0.7	92.11	Wil-Wor-Stur-Brat-Enf-Hart-NB-Ell-LM-Che-NH-West-Asso-Rev-Meth-Wil
		0.8	90.9	Wil-Meth-Rev-Asso-Wor-Stur-West-NH-LM-Brat-Enf-Che-NB-Hart-Ell-Wil
		0.85	**88.4**	**Wil-Stur-Wor-Brat-Enf-Che-NH-NB-Hart-Rev-Meth-Asso-West-LM-Ell-Wil**
		0.9	88.42	Wil-Hart-NB-NH-Che-Enf-Wor-Stur-Brat-Meth-Rev-Asso-Ell-LM-West-Wil
0.5				
	107.5	0.7	92.7	Wil-Che-NB-Ell-LM-Enf-West-Stur-Meth-Asso-Rev-Wor-NH-Hart-Brat-Wil
		0.8	89.59	Wil-Meth-Rev-Wor-Enf-Che-NH-Asso-NB-Hart-Ell-Stur-Brat-West-LM-Wil
		0.85	89.22	Wil-Asso-Rev-Meth-Hart-NB-Che-NH-LM-West-Brat-Enf-Wor-Stur-Ell-Wil
		0.9	92.11	Wil-Brat-Stur-Wor-Che-NB-Ell-Hart-Enf-West-LM-NH-Asso-Rev-Meth-Wil
0.7				
	102.1	0.7	89.59	Wil-Rev-Meth-Wor-Enf-NH-Che-Hart-NB-Asso-Stur-Ell-Brat-West-LM-Wil
		0.8	89.46	Wil-Meth-Rev-Asso-LM-Ell-Brat-NB-Hart-Che-NH-Enf-Wor-Stur-West-Wil
		0.85	**88.4**	**Wil-Stur-Wor-Brat-Enf-Che-NH-NB-Hart-Rev-Meth-Asso-West-LM-Ell-Wil**
		0.9	89.22	Wil-Asso-Rev-Meth-Hart-NB-Che-NH-LM-West-Brat-Enf-Wor-Stur-Ell-Wil
0.9				
	102.1	0.7	90.7	Wil-Brat-Ell-LM-NH-Che-NB-Hart-Enf-West-Stur-Asso-Meth-Rev-Wor-Wil
		0.8	91.43	Wil-Rev-Meth-Asso-Brat-Wor-Stur-NB-Che-NH-Enf-West-LM-Hart-Ell-Wil
		0.85	88.42	Wil-Brat-Stur-Wor-Enf-Hart-NH-Che-NB-Asso-Meth-Rev-Ell-LM-West-Wil
		0.9	90.47	Wil-Brat-Ell-West-Asso-Wor-LM-Enf-NH-Che-NB-Hart-Meth-Rev-Stur-Wil

Table 2. Comparative results obtained by using different initial solutions and cooling schedules

5.4 Sensitivity Analysis

Alhough the proposed SA heuristic solution procedure turned out to be useful for solving the real-world problem, its usefulness may be problem-specific. To further demonstrate its accuracy and robustness regardless of the problem setting, we performed a series of computational experiments followed by the sensitivity analysis. We started with these experiments by obtaining the exact optimal solution through the evaluations of all possible *n!* tours. The number of these tours is twice as many as that of the typical TSP tours due to asymmetric distances between customer nodes. For every tour, the TDD function under HOS regulations was used to calculate the travel time. The accuracy of the heuristic has been tested on different initial solutions obtained by different values of α. Values of α used in the experiments are 0.3, 0.5, 0.7, 0.9 and 1. The initial temperature is obtained by running M = 50 K (where K = 0.5 (n) (n-1)) experiments. The largest uphill step found in the iterations is set as the initial temperature and the lowest uphill step found is set as the halting criterion. $L = 0.5\ n^2$ iterations performed at each stage (temperature). Both exhaustive enumeration and SA heuristic are coded in JAVA using Eclipse 3.4. The computer programs were run on an Intel processor 2.0 GHz.

For the four-customer problems, both SA and Enumeration solved the problems using the almost same amount of computational time (4 to 6.5 seconds). For the six-customer problems, the computational time required by the enumeration method is 150 seconds, whereas the SA method required no more than 17 seconds. This comparison shows that for considerably small-sized problems (a maximum of up to six customers and a depot), SA produced optimal solutions. The major shortcoming of the enumeration method is that, with the increase in the number of customer nodes, its computational time increases exponentially, whereas SA's computational time increases polynomially. To prove that SA is a polynomial time algorithm, we conducted a series of experiments with different values of α and cooling schedules with up to nine customers and a depot. We also tested the proposed SA heuristic with 14, 19, 24 and 29 customers.

To elaborate, the computational time required to obtain the optimal solution using exhaustive enumeration for 10 nodes took more than a day, whereas the proposed SA took the average of 7 minutes to find an optimal solution or a near-optimal solution. Different initial solutions (different starting points in the solution space) are obtained using varying values of α, α = 0.3, 0.5, 0.7, 0.9. The objective value obtained by SA was compared to that obtained by the enumeration method. For each initial solution obtained by a given value of α, four sets of computational experiments were conducted using different cooling schedules with γ = 0.7, 0.8, 0.85, and 0.90. For a total of 16 experiments, we tried all the combinations of values of α and γ. In these experiments, we were able to locate the optimal solution about 70% of the time. The second best feasible solution was found by SA, when the optimal solution was not obtained (30% of the time). The deviation from the optimal objective value was below 1%. Also, with α =0.7 and cooling schedules with γ equal to 0.7, 0.8, 0.85 and 0.9, we consistently obtained optimal solutions. We found that the solution accuracy of SA mainly depend on the initial solution, initial temperature, number of iterations performed at each stage and the cooling schedule. Slower cooling requires more computational time but gives better quality solutions. Thus, by varying the cooling schedule, we can make trade-offs between the accuracy of solution and the computational time. By trial and error, we can determine the optimal cooling schedule for a problem with up to 10 customer nodes. The initial temperature was obtained by running M = 50 K (where K = 0.5 (n) ($n-1$)) experiments which required 4500 iterations. The largest uphill step found in the iterations is set as the initial temperature (δ_{max} = 180) and the lowest uphill step found is set as the halting criterion (δ_{min}=3). L = 0.5 n^2 iterations were performed at each stage (temperature).

Also, to assess the impact of time dependent arcs on total travel times, we constantly changed the departure time from the depot in an hour interval during the course of a day and plotted the optimal travel times in Figure 4. We discovered that a shortest travel time could be obtained when the departure time was 2 a.m. in the morning. The worst departure time appears to be around midnight. The difference between the longest and the shortest travel time is 38.22 hours, or 29% of the shortest travel time as shown in Figure 4. We noticed that the total travel time began to decrease sharply when the truck departed from the depot right after the midnight, while it began to increase when the truck left later in the morning or in the afternoon. A heavy traffic during the daytime caused a delay and thus a driver was likely to miss the delivery deadline. He/she must wait to begin service on the next day. Since the waiting time at the customer nodes could be impacted by rush hour traffic and restorative breaks during the trip, the sequence of customer visits might change. Thus, the optimal solution changes as the departure time changes during the day.

In Figure 5, we have plotted the average computational time required to solve the problem with a varying number of customers. The standard deviations with 10, 15, 20, 25 and 30 customers are 38.00, 50.77, 73.33, 104.86, 149.37 seconds, respectively. We have performed 10 iterations for each set of customers. All the experiments were performed with γ = 0.9 cooling schedule and L = 0.5(n^2) iterations at each stage, as suggested by Burkard and Rendl (1984). The initial temperature is obtained by running M = 50 K (where K = 0.5 (n) (n-1)) iterations. The number of iterations was increased by multiplying L by 1.03 in successive stages (temperatures). Through the experiments, we also observed that computational time increased polynomially as the number of customers increased. To find the causal relationship between the computational time and the number of customers visited, we developed a univariate regression model, $t = b\ n^c$ where b and c are positive constants and n is the number of customers. The regression analysis showed that the values of b and c were 2.506 and 2.257, respectively, or $t = 2.506 * n^{2.257}$. The fit is good as evidenced by the high R^2 value of 0.984.

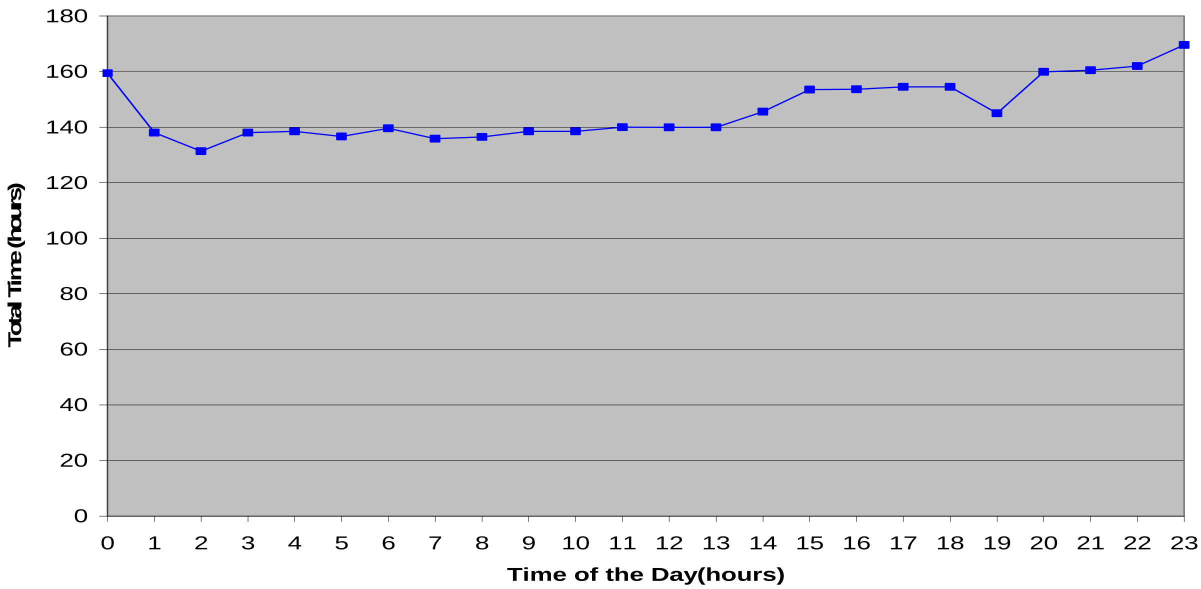

Figure 4. Travel time distribution

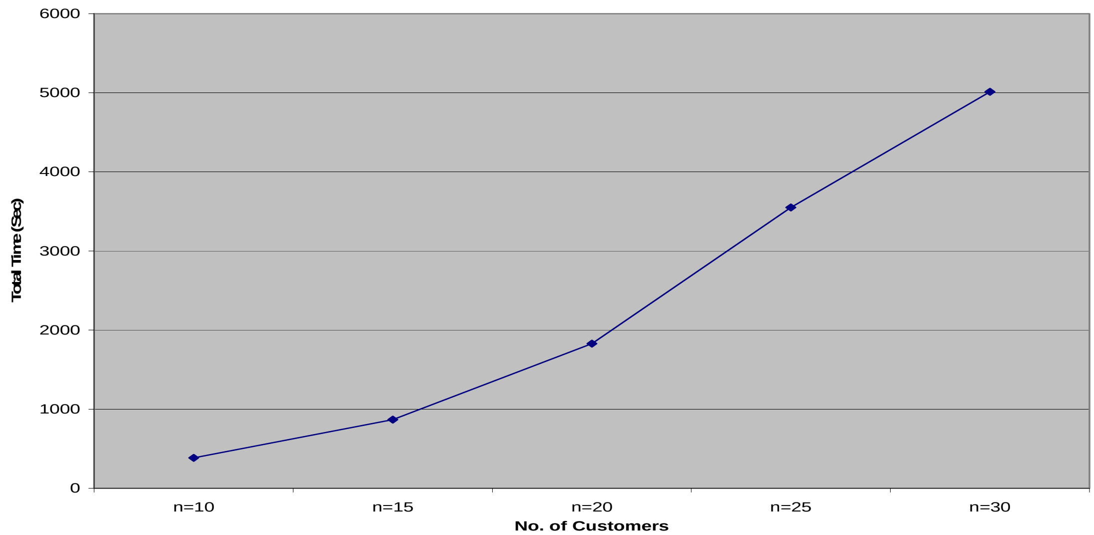

Figure 5. Computation time for larger sized problems

6 CONCLUDING REMARKS

This report proposes a simulated annealing meta-heuristic to tackle a time dependent, combined truck routing and driver scheduling problem under U.S. hours of service regulations. This problem was rarely addressed by the prior literature that overlooked driver safety issues resulting from extended driving hours and dynamics of hours of service regulations. In addition, the inherent computational complexity makes the problem hard to solve using the existing algorithm. This paper is one of the first attempts to solve such a problem using the alternative meta-heuristic called simulated annealing. The solution quality and computational efficiency of the proposed SA heuristic were verified by comparing its test results to those of the exhaustive enumeration algorithm. The test results were based on hypothetical but realistic problems encountered by a typical long-haul carrier in the U.S. The test results revealed that the enumeration method took more than one hour to solve the problem with 6 customers and more than a day to solve the problem with 9 customers. The comparative tests also show that the SA heuristic produced near-optimal solutions in a reasonable amount of time. Through a series of computational experiments, we found that the computational efficiency and accuracy of the proposed SA depended on the quality of its initial solution, equilibrium conditions, and the cooling schedule used to solve the problem. With that in mind, a series of computational experiments were conducted with different initial solutions, equilibrium conditions, and the cooling schedules. Based on these experiments, we discovered that a vast majority of the solutions obtained by SA were within 1 % of the optimal solutions. As a matter of fact, 75% of those solutions turned out to be optimal.

Despite the aforementioned proven efficiency and practicality, the proposed model and solution procedure point to a number of directions for future work:

- The model can be expanded to include the element of uncertainty (stochasticity) involved in the time windows as well as travel times between nodes.
- The model and solution procedure can be modified to consider the multiple objective aspect (e.g., trade-off between driver preferences and customer delivery requests) of the vehicle routing and scheduling problem similar to the one solved by Min (1991).
- Future research may accommodate delivery schedule changes dynamically through real time communication between the dispatcher and the drivers, i.e., for adding new delivery requests or canceling existing delivery requests.
- The comparison of the simulated annealing meta-heuristic to other proven heuristics such as ant-colony optimization, genetic algorithm, Tabu search, set partitioning, or *k*-opt exchange heuristics are worth investigating in the future.

Acknowledgments: The authors would like to thank the Intermodal Transport Institute at the University of Toledo and the U.S. Department of Transportation for funding this study through a research grant. Also, the authors are grateful to Mr. Ed Nagle, CEO of Nagle Trucking Inc. and Mr. Charles Harrett, CEO of Northern Continental Logistics for providing the authors with practical insights into hours-of-service regulations and their trucking operations in the United States.

REFERENCES

1. Archetti C., and Savelsbergh M. (2007), "The trip scheduling problem", Working paper, Georgia Institute of Technology, Atlanta, GA, www.isye.gatech.edu/~mwps/publications/.

2. Baker E.K. (1983), "An exact algorithm for the time constrained traveling salesman problem", *Operations Research*, 31, 939 – 945.

3. Bianco L., Mingozzi A., and Ricciardelli S. (1998),"Dynamic programming strategies for the traveling salesman problem with time windows and precedence constraints", *Operations Research,* 45, 365 – 378 .

4. Bodin, L, Golden, B, Assad, A, and Ball, M. (1983), "Routing and scheduling of vehicles and crews: The state of the art", *Computers and Operations Research*, 10, 62-212.

5. Breedam A.V. (1995), "Improvement Heuristics for the Vehicle Routing Problem based on Simulated Annealing", *European Journal of Operational Research*, 86, 480-490.

6. Brooks S.P. and Morgan B.J.T (1995), "Optimization using Simulated Annealing", *Operations Research*, 44, 241-257.

7. Burkard R.E., Rendl F. (1984), "A thermodynamically motivated simulation procedure for combinatorial optimization problems", *European Journal of Operational research*, 17, 169-174.

8. Calvo R.W. (2000), "A new heuristic for traveling salesman problem with time windows", *Transportation Science*, 34, 113-124.

9. Cerny V. (1985), "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm", *Journal of Optimization Theory and Applications*, 1, 41-52.

10. Clair L.A. and Fox S.D. (2004), "Time to Simplify Trucking Tariffs (Less Than Truckload)," *Supply Chain Management Review*, 8, 35-39.

11. Connolly D.T. (1990), “An improved annealing scheme for QAP”, *European Journal of Operational Research*, 46, 93-100.

12. Donati, A., Montemanni, R., Casagrande, N., Rizzoli, A., and Gambardella, L. (2006) “Time dependent vehicle routing problem with a multi ant colony system”, *European Journal of Operations Research*, 185, 1174- 1191.

13. Dumas, Y., Desrosiers, J. Gelinas, E., and Solomon, M. (1995), “An optimal algorithm for the traveling salesman problem with time windows,” *Operations Research*, 43, 367-371.

14. Eglese, R.W. (1990), “Simulated Annealing: A tool for Operations Research”, *European Journal of Operational Research*, 46, 271-281.

15. Federal Motor Carrier Safety Administration, (2005), *Hours-of-Service Regulations*, http://www.fmcsa.dot.gov/rules-regulations/topics/hos/hos-2005.htm.

16. Horne, J.A. and Reyner, L.A. (1995), “Sleep related vehicle accidents,” *BMJ Journal*, 310, 565-567.

17. Kirkpatrick, S., Gelatt Jr. C.D., and Vecchi, M.P. (1983), “Optimization by Simulated Annealing”, *Journal of Operations Research*, 220, 671-680.

18. Laarhoven, V.P.J.M and Aarts, E.H.L. (1987), *Simulated Annealing: Theory and Applications*, Kluwer Academic Publishers, Norwell, Massachusetts.

19. Laurent B., Guihaire V., and Hao J. (2005), “A heuristic solution for a driver-vehicle scheduling problem, Operations Research Proceedings”, Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Bremen, September 7–9.

20. Lowe, J. (2007), “Truck drivers’ new hours of service rules are struck down,” http://www.truckingaccidentattorneyblog.com/2007/10/truck_drivers_new_hours_of_ser_1.html.

21. Malandraki, C. and Daskin, M., (1992) “Time dependent vehicle routing problems: Formulations, Properties, and Heuristic Algorithms” *Transportation Science*, 26(33), 185-200.

22. Min, H. (1991), “A multiobjective vehicle routing problem with soft time windows: the case of public library distribution system,” *Socio-economic Planning Sciences*, 25(3), 179-188.

23. O’Hanlon, J.F. (1978), “What is the extent of the driver fatigue problem?” in *Driving fatigue in Road Traffic Accidents*, Brussel, Belgium: Commission of the European Communities Report No. EUR6065EN, pp. 19-25.

24. Osman, H.I. (1993) “Metastrategy simulated annealing and tabu search algorithms for vehicle routing problems”, *Annals of Operations Research*, 41, 421 – 451.

25. Portugal R., Ramalhinko H., and Paixao J.P., (2006) “Driver scheduling problem modeling”, Working Paper, Universida de de Lisboa, Portugal, http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1002565.

26. Reissman, C.J. (1997), *The Alert Driver: A Trucker’s Guide to Sleep, Fatigue, and Rest in our 24-Hour Society*, Alexandria, VA: American Trucking Associations, Inc.

27. Rene, V.V. (1993). Lecture Notes in Economics and Mathematical Systems: Applied simulated annealing, Berlin: Springer-Verlag.

28. Solomon, M and Desrosiers, J. (1988), “Time window constrained routing and scheduling problems”, *Transportation Science*, 22, 1-13.

29. Soumia I., Michel G. and Jean-Yves, P. (2003), “Vehicle dispatching with time-dependent travel times”, *European Journal of Operational* Research, 144, 379–396.

30. Timothy G. (2007), “An operating room scheduling problem solved using simulated annealing”, MS Thesis, Dalhousie University, Canada.

31. WERC (2004), “Under the Gun: As the New Trucking Hours of Service rules take the hold, is the Warehousing Industry ready?” *WERC Sheet*, March, 1-3.

32. Xu, H., Chen, Z.L., Rajagopal, S., and Arunapuram, S. (2003), “Solving a practical pickup and delivery problem”, *Transportation Science*, 37(3), 347-364.

Appendix A. Arcs of the Network

Arc's origin node	Arc's destination node	Distance (miles)
Methuen	I495_I93_via_I495	16
I495_I93_via_I495	Methuen	16
Methuen	I495_I93_via_I93	9
I495_I93_via_I93	Methuen	9
I495_I93_via_I93	I93_I95	19
I93_I95	I495_I93_via_I93	19
I93_I95	Revere	24
Revere	I93_I95	24
Revere	BOS	15
BOS	Revere	15
I93_I95	BOS	20
BOS	I93_I95	20
I93_I95	R2_I95	17
R2_I95	I93_I95	17
R2_I95	R2_I495	24
R2_I495	R2_I95	24
I495_I93_via_I495	R2_I495	31
R2_I495	I495_I93_via_I495	31
R2_I95	I90_I95	10
I90_I95	R2_I95	10
I90_I95	R1_I95	20
R1_I95	I90_I95	20
R1_I95	I93_R24	5
I93_R24	R1_I95	5
I93_R24	BOS	20
BOS	I93_R24	20
BOS	I90_I95	17
I90_I95	BOS	17
I90_I95	I90_I495	27
I90_I495	I90_I95	27
R2_I495	I290_I495	20
I290_I495	R2_I495	20
I290_I495	I90_I495	11
I90_I495	I290_I495	11
I90_I495	I95_I495	40
I95_I495	I90_I495	40

Arc’s origin node	Arc’s destination node	Distance (miles)
R1_I95	I95_I495	24
I95_I4I95	R1_I95	24
I95_I4I95	I495_R24	22
I495_R24	I95_I495	22
I495_R24	I93_R24	29
I93_R24	I495_R24	29
I95_I495	PRVD	30
PRVD	I95_I495	30
PRVD	Assonet	42
Assonet	PRVD	42
Assonet	I495_R24	18
I495_R24	Assonet	18
PRVD	I90_I146	58
I90_I146	PRVD	58
I90_I146	I90_I495	20
I90_I495	I90_I146	20
I90_I395	I90_I146	6
I90_I146	I90_I395	6
I90_I395	Worcester	10
Worcester	I90_I395	10
Worcester	I290_I495	22
I290_I495	Worcester	22
Worcester	I190_R2	33
I190_R2	Worcester	33
I190_R2	R2_I495	19
R2_I495	I190_R2	19
I84_I83	Hartford	18
Hartford	I84_I83	18
Westfield	I90_I91	9
I90_I91	Westfield	9
I91_R5	Enfield	30
Enfield	I91_R5	30
Enfield	Hartford	26
Hartford	Enfield	26
Hartford	I91_R9	18
I91_R9	Hartford	18
Hartford	R6_R9	12
R6_R9	Hartford	12

Arc’s origin node	Arc’s destination node	Distance (miles)
R6_R9	New_Britain	8
New_Britain	R6_R9	8
R6_R9	I84_I72	10
I84_I72	R6_R9	10
New_Britain	I84_I72	5
I84_I72	New_Britain	5
New_Britain	I91_R9	11
I91_R9	New_Britain	11
I91_R9	I91_I691	12
I91_I691	I91_R9	12
I91_I691	NHV	30
NHV	I91_I691	30
NHV	New_Haven	11
New_Haven	NHV	11
Worcester	I90_I146	8
I90_I146	Worcester	8
I90_I395	Sturbridge	25
Sturbridge	I90_I395	25
Sturbridge	I84_I83	53
I84_I83	Sturbridge	53
Sturbridge	I90_R32	30
I90_R32	Sturbridge	30
I90_R32	I90_R21	15
I90_R21	I90_R32	15
I90_R21	I90_I91	14
I90_I91	I90_R21	14
I90_I91	I91_R5	11
I91_R5	I90_I91	11
I90_R32	Wilbraham	13
Wilbraham	I90_R32	13
Wilbraham	I90_R21	9
I90_R21	Wilbraham	9
Wilbraham	I83_H_road	13
I83_H_road	Wilbraham	13
I83_H_road	Ellington	11
Ellington	I83_H_road	11
Ellington	I84_I83	11
I84_I83	Ellington	11
I83_H_road	Long_Meadow	8

Arc's origin node	Arc's destination node	Distance (miles)
Long_Meadow	I83_H_road	8
Long_Meadow	I91_R5	8
I91_R5	Long_Meadow	8
PRVD	R1_I395	95
R1_I395	PRVD	95
I90_I395	R1_I395	110
R1_I395	I90_I395	110
R1_I395	I91_R9	64
I91_R9	R1_I395	64
R1_I395	New_Haven	66
New_Haven	R1_I395	66
I84_I72	R10_I691	18
R10_I691	I84_I72	18
R10_I691	I91_I691	11
I91_I691	R10_I691	11
R10_I691	Cheshire	6
Cheshire	R10_I691	6
Cheshire	NHV	15
NHV	Cheshire	15
I90_I91	I91_R2	56
I91_R2	I90_I91	56
I91_R2	Brattleboro	32
Brattleboro	I91_R2	32
I91_R2	I190_R2	82
I190_R2	I91_R2	82

Appendix B
Time Dependent Speed of the Arcs of the Network

Arc	Speed (miles/hour) for each hour of the day starting at midnight
MethuenI495_I93_via_I495	65 65 65 62 60 60 60 58 58 60 60 60 60 60 60 60 60 60 60 60 60 62 65 65
I495_I93_via_I495Methuen	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 56 61 65 65
MethuenI495_I93_via_I93	65 65 65 63 60 60 60 60 60 60 60 58 60 60 60 60 60 60 60 60 60 63 65 65
I495_I93_via_I93Methuen	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 50 40 60 60 60 63 65 65
I495_I93_via_I93I93_I95	65 65 65 63 60 60 56 53 60 60 58 60 60 60 60 60 60 60 60 60 60 63 65 65
I93_I95I495_I93_via_I93	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I93_I95Revere	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
RevereI93_I95	65 65 65 63 60 60 60 50 53 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
RevereBOS	65 65 65 63 60 60 60 52 55 60 60 60 60 60 60 60 60 60 60 60 55 60 65 65
BOSRevere	65 65 65 63 60 60 60 60 60 60 60 60 60 60 57 57 56 60 60 60 60 63 65 65
I93_I95BOS	65 65 65 63 60 55 40 30 30 47 55 60 60 60 60 60 60 60 60 60 60 63 65 65
BOSI93_I95	65 65 65 63 60 60 60 53 53 60 60 60 60 60 57 47 40 47 47 57 60 63 65 65
I93_I95R2_I95	65 65 65 63 60 60 60 48 47 58 60 60 60 60 60 58 58 58 60 60 60 63 65 65
R2_I95I93_I95	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 55 43 40 60 60 60 63 65 65
R2_I95R2_I495	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
R2_I495R2_I95	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I495_I93_via_I495R2_I495	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
R2_I495I495_I93_via_I495	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
R2_I95I90_I95	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I90_I95R2_I95	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I90_I95R1_I95	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
R1_I95I90_I95	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
R1_I95I93_R24	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 56 40 40 54 60 60 63 65 65
I93_R24R1_I95	65 65 65 63 60 60 60 57 58 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I93_R24BOS	65 65 65 63 60 60 44 40 47 50 60 60 60 60 55 55 54 54 55 60 60 63 65 65
BOSI93_R24	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 53 53 54 54 60 60 63 65 65
BOSI90_I95	65 65 65 63 60 60 55 60 60 60 60 60 60 60 60 60 55 53 60 60 60 63 65 65
I90_I95BOS	65 65 65 63 60 60 55 56 56 60 60 60 60 60 60 60 60 52 54 60 60 63 65 65
I90_I95I90_I495	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 58 58 58 60 60 63 65 65
I90_I495I90_I95	65 65 65 63 60 60 57 50 48 54 60 60 60 60 60 60 60 60 60 60 60 63 65 65
R2_I495I290_I495	65 65 65 63 60 60 60 60 60 60 56 53 60 60 60 60 60 60 60 60 60 63 65 65
I290_I495R2_I495	65 65 65 63 60 60 60 60 60 58 60 58 60 60 60 60 60 60 60 60 58 62 65 65
I290_I495I90_I495	65 65 65 61 58 60 60 60 60 60 54 60 60 60 60 60 60 57 57 50 52 59 65 65
I90_I495I290_I495	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I90_I495I95_I495	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 58 61 65 65
I95_I495I90_I495	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
R1_I95I95_I495	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 57 57 60 60 60 63 65 65
Arc	**Speed (miles/hour) for each hour of the day starting at midnight**

I95_I4I95R1_I95	65 65 65 63 60 60 57 48 52 60 60 60 60 60 60 60 60 57 57 60 60 63 65 65
I95_I4I95I495_R24	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 58 54 55 58 62 65 65
I495_R24I95_I495	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I495_R24I93_R24	65 65 65 63 60 60 60 50 52 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I93_R24I495_R24	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 58 60 60 60 60 63 65 65
I95_I495PRVD	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 58 60 60 60 60 63 65 65
PRVDI95_I495	65 65 65 63 60 60 60 60 60 60 60 60 58 60 60 60 60 60 60 60 60 63 65 65
PRVDAssonet	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 58 60 60 57 57 61 65 65
AssonetPRVD	65 65 65 63 60 60 60 57 57 60 60 60 60 60 60 60 53 60 60 56 56 60 65 65
AssonetI495_R24	65 65 65 63 60 60 60 50 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I495_R24Assonet	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 57 56 60 60 63 65 65
PRVDI90_I146	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 58 60 63 65 65
I90_I146PRVD	65 65 65 63 60 60 60 59 60 60 60 60 60 60 60 60 60 60 58 60 56 60 65 65
I90_I146I90_I495	65 65 65 63 60 60 60 60 58 56 56 58 60 60 60 60 60 60 60 60 60 63 65 65
I90_I495I90_I146	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I90_I395I90_I146	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I90_I146I90_I395	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I90_I395Worcester	65 65 65 63 60 60 60 53 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
WorcesterI90_I395	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
WorcesterI290_I495	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I290_I495Worcester	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
WorcesterI190_R2	65 65 65 63 60 60 60 60 60 58 58 59 60 60 60 60 59 60 60 60 58 62 65 65
I190_R2Worcester	65 65 65 60 55 60 60 60 60 60 60 60 60 60 60 60 60 58 60 60 60 63 65 65
I190_R2R2_I495	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
R2_I495I190_R2	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I84_I83Hartford	65 65 65 63 60 60 58 55 41 50 60 60 60 60 60 57 52 60 60 60 60 63 65 65
HartfordI84_I83	65 65 65 63 60 60 60 60 60 60 56 60 60 57 60 60 60 52 60 60 60 63 65 65
WestfieldI90_I91	65 65 65 63 60 60 60 51 58 58 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I90_I91Westfield	65 65 65 63 60 60 60 60 60 58 60 60 60 60 58 44 53 60 60 60 60 63 65 65
I91_R5Enfield	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
EnfieldI91_R5	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 57 49 50 41 60 63 65 65
EnfieldHartford	65 65 65 63 60 60 58 50 57 60 60 60 60 60 60 57 51 50 55 60 60 63 65 65
HartfordEnfield	65 65 65 63 60 60 60 60 58 60 60 60 60 60 60 60 58 60 60 60 60 63 65 65
HartfordI91_R9	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 58 60 57 57 60 63 65 65
I91_R9Hartford	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 57 57 60 57 60 60 63 65 65
HartfordR6_R9	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
R6_R9Hartford	65 65 65 62 58 60 60 50 60 60 60 57 60 60 60 57 49 45 56 60 60 63 65 65
R6_R9New_Britain	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 57 60 60 60 47 50 58 65 65
New_BritainR6_R9	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
R6_R9I84_I72	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65

Arc	Speed (miles/hour) for each hour of the day starting at midnight
I84_I72R6_R9	65 65 65 63 60 60 58 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
New_BritainI84_I72	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I84_I72New_Britain	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 54 60 63 65 65
New_BritainI91_R9	65 65 65 63 60 60 60 58 60 60 60 60 56 58 58 60 60 60 60 60 47 56 65 65
I91_R9New_Britain	65 65 65 63 60 60 60 60 53 60 60 56 57 60 60 60 60 50 60 60 60 63 65 65
I91_R9I91_I691	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 51 58 65 65
I91_I691I91_R9	65 65 65 63 60 60 60 60 60 60 60 60 56 60 60 60 60 60 56 60 60 63 65 65
I91_I691NHV	65 65 65 63 60 60 60 60 60 60 57 60 60 60 58 54 60 60 60 60 60 63 65 65
NHVI91_I691	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 58 60 63 65 65
NHVNew_Haven	65 65 65 63 60 60 60 53 50 60 60 60 60 60 60 55 53 54 60 60 60 63 65 65
New_HavenNHV	65 65 65 61 57 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
WorcesterI90_I146	63 63 63 60 58 58 58 56 58 58 58 58 58 58 58 58 58 58 58 58 58 60 63 63
I90_I146Worcester	63 63 63 60 58 58 58 56 58 58 58 58 58 58 58 58 58 58 58 58 58 60 63 63
I90_I395Sturbridge	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
SturbridgeI90_I395	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
SturbridgeI84_I83	65 65 65 63 60 60 60 60 60 60 57 60 60 60 60 60 60 60 58 60 60 63 65 65
I84_I83Sturbridge	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 57 60 60 60 63 65 65
SturbridgeI90_R32	65 65 65 63 60 60 60 60 60 60 60 57 58 58 55 60 60 60 55 53 60 63 65 65
I90_R32Sturbridge	65 65 65 63 60 60 60 56 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I90_R32I90_R21	65 65 65 63 60 60 60 60 60 50 58 52 60 60 60 60 60 57 56 57 60 63 65 65
I90_R21I90_R32	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 52 60 60 63 65 65
I90_R21I90_I91	65 65 65 63 60 60 60 60 60 56 52 57 60 60 60 60 60 60 60 60 60 63 65 65
I90_I91I90_R21	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I90_I91I91_R5	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 57 60 60 60 63 65 65
I91_R5I90_I91	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I90_R32Wilbraham	60 60 60 58 55 55 55 55 55 53 55 55 55 55 55 55 55 53 53 55 55 58 60 60
WilbrahamI90_R32	60 60 60 58 55 55 55 55 55 53 55 55 55 55 55 55 55 53 53 55 55 58 60 60
WilbrahamI90_R21	60 60 60 58 55 55 55 55 55 52 53 54 55 55 55 55 55 55 55 54 54 57 60 60
I90_R21Wilbraham	60 60 60 58 55 55 55 55 55 52 53 54 55 55 55 55 55 55 55 54 54 57 60 60
WilbrahamI83_H_road	50 50 50 48 45 45 44 43 43 45 45 45 45 45 45 45 43 43 44 45 45 48 50 50
I83_H_roadWilbraham	50 50 50 48 45 45 44 43 43 45 45 45 45 45 45 45 43 43 44 45 45 48 50 50
I83_H_roadEllington	55 55 55 53 50 50 49 48 48 50 50 50 50 50 50 50 48 48 49 50 50 53 55 55
EllingtonI83_H_road	55 55 55 53 50 50 49 48 48 50 50 50 50 50 50 50 48 48 49 50 50 53 55 55
EllingtonI84_I83	55 55 55 53 50 50 49 48 48 50 50 50 50 50 50 50 48 48 49 50 50 53 55 55
I84_I83Ellington	55 55 55 53 50 50 49 48 48 50 50 50 50 50 50 50 48 48 49 50 50 53 55 55
I83_H_roadLong_Meadow	55 55 55 53 50 50 49 48 48 50 50 50 50 50 50 50 48 48 49 50 50 53 55 55
Long_MeadowI83_H_road	55 55 55 53 50 50 49 48 48 50 50 50 50 50 50 50 48 48 49 50 50 53 55 55
Long_MeadowI91_R5	55 55 55 53 50 50 49 48 48 50 50 50 50 50 50 50 48 48 49 50 50 53 55 55
I91_R5Long_Meadow	55 55 55 53 50 50 49 48 48 50 50 50 50 50 50 50 48 48 49 50 50 53 55 55

PRVDR1_I395	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
Arc	**Speed (miles/hour) for each hour of the day starting at midnight**
R1_I395PRVD	65 65 65 63 60 60 60 60 59 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
I90_I395R1_I395	65 65 65 62 59 60 60 60 60 60 60 60 60 60 58 60 60 60 60 59 60 63 65 65
R1_I395I90_I395	65 65 65 61 58 57 57 59 60 60 60 58 60 60 60 60 59 60 60 59 50 58 65 65
R1_I395I91_R9	65 65 65 62 59 60 60 60 60 60 60 60 60 60 58 60 60 60 60 60 58 61 65 65
I91_R9R1_I395	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 58 58 60 60 60 63 65 65
R1_I395New_Haven	65 65 65 62 59 60 60 57 58 60 56 58 60 58 58 60 60 60 60 60 60 63 65 65
New_HavenR1_I395	65 65 65 63 60 60 60 60 60 60 60 58 60 60 60 60 60 60 60 60 60 63 65 65
I84_I72R10_I691	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
R10_I691I84_I72	65 65 65 63 60 60 60 60 60 60 60 60 60 60 58 60 60 60 60 60 60 63 65 65
R10_I691I91_I691	65 65 65 63 60 60 60 57 60 60 60 60 60 60 60 60 60 60 53 58 60 63 65 65
I91_I691R10_I691	65 65 65 63 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 63 65 65
R10_I691Cheshire	55 55 55 53 50 50 50 48 50 50 50 50 50 50 50 50 50 50 48 49 50 53 55 55
CheshireR10_I691	55 55 55 53 50 50 50 48 50 50 50 50 50 50 50 50 50 50 48 49 50 53 55 55
CheshireNHV	55 55 55 53 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 53 55 55
NHVCheshire	55 55 55 53 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 53 55 55
I90_I91I91_R2	65 65 65 63 60 60 60 60 60 60 60 59 59 59 57 60 57 56 60 60 60 63 65 65
I91_R2I90_I91	65 65 65 61 57 60 60 60 59 60 60 60 60 60 60 60 60 60 60 58 60 63 65 65
I91_R2Brattleboro	65 65 65 63 60 60 60 60 59 60 60 60 60 60 60 60 57 56 60 60 60 63 65 65
BrattleboroI91_R2	65 65 65 63 60 60 60 60 59 60 60 60 60 60 60 60 57 56 60 60 60 63 65 65
I91_R2I190_R2	55 55 55 53 50 50 50 50 50 50 50 50 50 50 50 50 49 48 50 50 50 53 55 55
I190_R2I91_R2	55 55 55 53 50 50 50 50 50 50 50 50 50 50 50 50 49 48 50 50 50 53 55 55

Appendix C
Truck Route and Driver’s Schedule

Truck leaves **Wilbraham depot** at 7.00 (7:00 am of the first day)

Wilbraham to I90_R32 to Sturbridge (0.74 hours)
Time of arrival at **Sturbridge** customer: 7.74
Truck **waits** until 9:00 to start service; service lasts 2 hours
Time of departure from Sturbridge to the customer at Worcester: 11.00

Sturbridge to I90_I395 to Worcester (0.58 hours)
Time of arrival at **Worcester** customer: 11.58
Service lasts 2 hours
Time of departure from Worcester to the customer at Brattleboro:13.58

Worcester to I190_R2 to I91_R2 to Brattleboro (2.74 hours)
Time of arrival at **Brattleboro** customer:16.32
Service lasts 2 hours
Time of departure from Brattleboro to Enfield (home of the driver):18.32

Brattleboro to I91_R2 to I90_I91 to I91_R5 to Enfield (2.18 hours)
Time of arrival at **Enfield (home of the driver)**: 20.50 (8:30 pm)
Driver stays overnight at home until 7:00 am and takes **off-duty break**.
Time of departure from Enfield toward the customer at Cheshire: 31.00

Enfield to Hartford to R6_R9 to I84_I72 to R10_I691 to Cheshire (1.3 hours)
Time of arrival at **Cheshire** customer: 32.31
Truck **waits** until 9:00 am to start service; service lasts 2 hours
Time of departure from Cheshire to the customer at New Haven: 35.0

Cheshire to NHV to New_Haven (0.48 hours)
Time of arrival at **New_Haven** customer: 35.48
Service lasts 2 hours
Time of departure from New Haven to the customer at New Britain: 37.48

New_Haven to NHV to I91_I691 to I91_R9 to New_Britain (1.07 hours)
Time of arrival at **New_Britain** customer: 38.55
Service lasts 2 hours
Time of Departure from New_Britain to the customer at Hartford: 40.55

New_Britain to R6_R9 to Hartford (0.38 hours)
Time of arrival at **Hartford** customer: 40.93
Service lasts 2 hours
Time of Departure from Hartford to the customer at Revere: 42.93

Hartford to I84_I83 to Sturbridge to I90_I395 to I90_I146 to I90_I495 to I90_I95 to BOS to Revere (2.99 hours)
Time of arrival at customer: 45.92
Driver stays overnight at customer taking an **off-duty break**.
Truck **waits** until 9:00 am to start service; service lasts 2 hours
Time of Departure from Revere to the customer at Methuen: 59.0

Revere to I93_I95 to I495_I93_via_I93 to Methuen (0.87 hours)
Time of arrival at **Methuen** customer: 59.87
Service lasts 2 hours
Time of Departure from Methuen to the customer at Assonet is: 61.87

Methuen to I495_I93_via_I93 to I93_I95 to BOS to I93_R24 to I495_R24 to Assonet (1.91 hours)

Time of arrival at **Assonet** customer: 63.78
Service lasts 2 hours
Time of Departure from Assonet to the customer at Westfield: 65.78

Assonet to PRVD to I90_I146 to I90_I395 to Sturbridge to I90_R32 to I90_R21 to I90_I91 to Westfield (3.33 hours)
Time of arrival at **Westfield** customer: 69.11
Truck arrives late (9.11 pm) for service.
Driver stays overnight at customer taking an **off-duty break**.
Truck **waits** until 9:00 am to start service; service lasts 2 hours.
Time of Departure from Westfield to the customer at Long_Meadow: 83.0

Westfield to I90_I91 to I91_R5 to Long_Meadow (0.49 hours)
Time of arrival at **Long_Meadow** customer: 83.49
Service lasts 2 hours
Time of Departure from Long_Meadow to the customer at Ellington: 85.49

Long_Meadow to I83_H_road to Ellington (0.38 hours)
Time of arrival at **Ellington** customer: 85.87
Service lasts 2 hours
Time of Departure from Ellington to the customer at Wilbraham: 87.87

Highway route traversed: Ellington to I83_H_road to Wilbraham
Time of arrival at **Wilbraham depot**: 88.40 (4:40 pm of the fourth day