

TO: Professor Don Graham, PhD.

FROM: John Hokkanen, Predict 490

DATE: August 12, 2017

RE: Estimation of Freight Transportation Miles

Purpose of the Report

To assess proposals from external common carriers, Northeastern Home Goods (“NHG” or “the Company”) has requested an analysis and estimate of routing delivery miles based on historic freight data. To properly assess the proposals, the Company wishes to understand what is required to achieve the following for its distribution system: 1) maintain its fixed daily delivery schedule of shipping from a central distribution center (“DC”); 2) maximize container volumes for trucks and minimize mileage to reduce costs; and 3) maintain all mandatory DOT requirements. Understanding how these various goals interact with each other is central to the proper assessment of proposals that the Company receives.

Executive Summary

The total cost of shipping is determined by the cost of equipment (primarily trucks and trailers, but also licensing, maintenance and storage), truck driving labor, and mileage (i.e., fuel, taxes). However, the analysis below was performed with a sole goal of minimizing total mileage. Using this common metric will allow for easy comparison of third party common carrier proposals. Estimates for optimized aggregate miles for a representative week are as follows:

Day	Week Total	Mon	Tue	Wed	Thu	Fri
Trucks	31	6	5	7	7	6
Volume	65429	10223	11537	15192	15009	13468
Miles	6638	1388 ⁴	1370	1196	1446	1238
Labor³	254	52	55	35	58	53
ON Mins¹	538.5	37.5	87	124.5	121.5	168
Cost²	\$10,995	\$2,272	\$2,340	\$1,719	\$2,472	\$2,192
\$/100ft³	\$16.80	\$22.22	\$20.28	\$11.31	\$16.47	\$16.28

¹ON Mins means minutes of overnight travel. The number in the parentheses indicates the number of trucks that had routes that required overnight travel.

²Cost was calculated at a rate of 70 cents/mile and \$25/hour for labor.

³Labor costs are in hours of driving; these do not include resting time. See ON Mins if a salary surcharge applies to overnight travel.

⁴After completion of the study, an additional run produced **an even lower Monday distance of 1355**. It is noted here but not included in the totals. The route is presented in the raw routing data in the last appendix.

Resource Requirements

As the table shows, an average week typically requires either six or seven trucks and drivers per day. (See below for discussion of truck requirements.) All five days required an overnight

sleeper cab for one route, though the aggregate overtime minutes totaled less than nine hours of driving that could not be accomplished during the five fourteen hour workdays of multiple vehicles/drivers. Fewer overnight runs could be constructed if necessary (e.g., due to labor agreements) as our software is able to trade off a few additional miles to eliminate an overnight route. The cost estimates are intended for relative comparison, but they reveal a few insights:

- Higher volumes typically have a lower cost per volume. For example, the labor and mileage cost for 100 ft³ on Monday is almost double the cost of Wednesday. That result does not come as a surprise as optimization for spatial relatedness and vehicle fit offers larger quantities to benefit from economies of scale.
- Minimizing mileage reduces both labor and mileage costs at about the same rate. For example, a reduction of forty miles reduces labor by \$25 and mileage expense by \$28. By optimizing for just mileage, Company costs will be reduced significantly.
- Optimizing for multiple deliveries to the same location is essential for containing costs because additional deliveries incur no mileage cost and, usually, no additional labor cost. In effect, optimizing for multiple deliveries yields free deliveries.
- With a high-quality routing application, there may be circumstances when it is beneficial to break up large (i.e., >1000 ft³) orders. Doing so would allow the system to save money by balancing the GIS-relatedness of orders with the ability to pack the trucks more fully while simultaneously considering the labor costs of unloading.

The study optimized for different truck amounts and shows the basic tradeoff between fewest miles and fewest trucks:

	Fewest Miles		Fewest Trucks	
	Miles	Trucks	Miles	Trucks
Mon	1388	6	1494	5
Tue	1370	5	1370	5
Wed	1196	7	1481	6
Thu	1446	7	1520	6
Fri	1238	6	1561	5
Sum	6638		7426	
Year				
Miles	345,176		386,152	

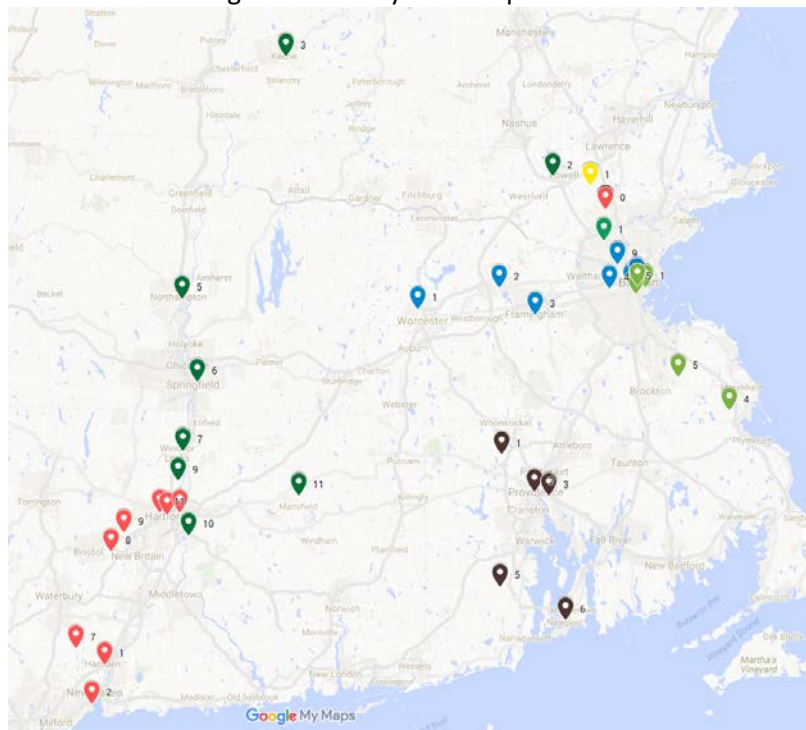
As the table above shows, the use of an optimizing algorithm with a flexible number of trucks can save the company money by allowing the company to balance these tradeoffs and costs for fewer miles and fewer trucks. In general, using the absolute least number of trucks will result in higher mileage, especially at larger order volumes; with additional trucks, there is more freedom to create a new optimal route which enhances the efficiency of other routes. Experimental results show that the specifics of the orders will determine the relationship between trucks and miles. Sometimes, using fewer trucks results in fewer miles because the GIS-related orders can be placed in the same truck. However, when the spatially related orders cannot all fit in the same truck, adding trucks can allow a

reduction of miles because added capacity allows for related packages to be grouped together thus lowering total miles. In other words, the highest level of efficiency occurs when trucks only receive orders which are spatially related in a sensible route, taking into account the available space of the truck.

At the same time, if you add capacity beyond what is required to group the spatially-related packages in a route, then a decrease in efficiency will occur as multiple trucks are duplicating portions of routes and thus increase total mileage. Finally, reducing truck counts to the minimum required by volume is not always possible because some order volume and location combinations will not allow one to have a valid route plan even though the specified trucks have adequate room. The driver is unable to perform all of the driving and unloading to all of the destinations by the end of the day even though the vehicle has the capacity to carry the freight. In this latter situation, the only solution to obtain a valid route plan is to add more trucks.

A quick visual inspection of a day's routes quickly shows the utility of the prototype scheduler. The map image below displays the routes of an average Wednesday. (Detailed routes for all days are set forth in Appendix C.) In this map each vehicle route is plotted with a different color pushpin, with the adjacent numbers showing the stop number for that truck. As the image shows, the system has clustered spatially-related stops in a single vehicle route thereby minimizing the total travel distance. For an interactive Google Map of all routes/stops in the sample routing report, visit the specified link.

Average Wednesday Route Optimized Plan



<https://drive.google.com/open?id=1ZIG9GC2uQ1YgAaEkzcNTTcVQajY&usp=sharing>

Based on this study, the Company has the expertise and tools to review third party proposals and partner with a common carrier. Alternatively, this capability could be developed further and the logistics function could be managed internally. When requesting proposals, the Company should request that all proposals include a sample routing schedule based on the data used in this study. Doing so will enable the Company to fairly assess the third party's ability to execute a well-designed route and contain our Company's costs. In addition, the Company might request carriage-only pricing where the third party only does the driving.

Detailed Description of the Study

Functional Requirements for the Project

The Company's requirements for routing may be summarized as follows:

- 1) All orders are on a fixed delivery schedule from the distribution center.
- 2) All vehicles will commence first delivery at 8:00am each day for their route and be completely unloaded no later than 6:00pm on the day that the truck leaves the distribution center.
- 3) All mandatory DOT regulations are to be followed. Specifically, a driver may not drive more than 11 hours within a 14-hour workday or work past the workday. If the driver achieves either threshold (i.e., the 11 hour drive threshold or the 14 hour workday threshold), then driver must rest for 10 hours before continuing. The combination of these conditions with the preceding requirement means that drivers must complete all deliveries and be on the last leg of the return trip home before either threshold is met.
- 4) Travel times are calculated on a 1.5 minute per mile traveled basis.
- 5) Unload times are calculated at .03 per ft³, with a minimum 30 minute minimum. However, multiple order deliveries to the same location at the same time may be aggregated for purposes of unload times.
- 6) All vehicles have a maximum container volume of 3200 ft³.
- 7) Sleeper cabs are available where overnight trips are required.
- 8) As many trucks as are necessarily for optimal delivery will be available.
- 9) Routes are to be optimized solely to minimize total mileage; it will be the responsibility of the common carriers to manage truck counts and types, labor issues and overnight stays.

Formal Problem Formulation

The foregoing business rules may be translated into a more precise mathematical notation for precision in understanding and system design.

Problem Description

This problem consists of a set of vehicles V , and a set of orders O represented by a network of nodes N (order delivery locations) connected by arcs A which represent the distance between order delivery locations. The goal is to visit all delivery locations, i.e., deliver all orders, so that the objective is met while simultaneously conforming to the constraints.

Objective: Minimize total distance

Constraints:

- Each order is delivered once
- Vehicle capacity constraints are observed
- Time window constraints are observed
- DOT operational constraints are observed
- Each route starts and ends at the distribution center

Variable Definition

These definitions exceed formal requirements for the formulation of the constraints, and calculations are declared where possible for clarity and implementation. Please note a few variables may only appear in calculations and not in the problem formulation.

n is the total order count for all vehicles, which is the same as the number of order delivery locations

k is the vehicle count, $k \leq n$ (i.e., no empty trucks)

0 represents the starting distribution center node

$n+1$ represents the distribution center node to which vehicles return

$O = \{1, 2, \dots, n\}$ orders; Order set O represents the set of orders to be delivered to the stores,

$N = \{0, 1, \dots, n, n+1\}$ node size; Set N comprises the complete set of nodes from distribution starting point through order delivery nodes to distribution center return location.

$K = \{1, 2, \dots, k\}$; Set K represents the set of vehicles.

Arc set A represents all possible connections between nodes in the network

c_{ij} = arc distance (cost) between order locations i and j . Note: Order locations are not required to be unique. Thus, where two or more order delivery locations are identical, the arc distance cost between them will be zero.

$x_{ijk} = 1$ if vehicle k drives from node i to j , and 0 otherwise, $\forall i \in N, \forall j \in N, \forall k \in K$

s_{ij} = average minutes per mile between order locations i and j , $i \in N, j \in N$. Currently $s_{ij} = 1.5$ minutes per mile, $\forall i \in N, \forall j \in N$

*t_{ij} = travel time in minutes from node i to j , $i \in N, j \in N$. $t_{ij} = c_{ij} * s_{ij}$, $\forall i \in N, \forall j \in N$*

q_k = capacity of vehicle k , currently $q_k = 3200 \text{ ft}^3$, $\forall k \in K$

v_i = volume of order i to be delivered to node i , $i \in O$

$w_i = 1$ if volume v_i is ≥ 1000 , and 0 otherwise, for order i at node i , $i \in O$

u_i = unload time in minutes for order i at node i , $i \in O$,

the general form without accounting for multiple orders may be calculated as follows:

$$u_{ik} = 30 + w_{ik}(.03 * v_i - 30), \forall i \in O, \forall k \in K$$

o_i = opening time of at delivery location i , currently $o_i = 8 * 60, \forall i \in O$

z_i = closing time of at delivery location i , currently $z_i = 18 * 60, \forall i \in O$

d_{ik} = time in minutes that vehicle k departs from node i for its next destination,

$i \in \{0, 1, 2, \dots, n\}, \forall k \in K$, which may be calculated as follows:

$d_{0k} = \sum_{j \in O} x_{0jk} (o_j - t_{0j}), \forall k \in K$ (departure from the distribution center),

$d_{ik} = \sum_{j \in O} x_{ijk} (a_{ik} + u_{ik}), \forall i \in O, \forall k \in K$ (departure from order delivery locations)

$d_{n+1,k}$ is always undefined and has no meaning because the return to the distribution center has no departure.

a_{ik} = time in minutes that vehicle k arrives at to deliver order $i, \forall i \in N, \forall k \in K$, which may be calculated as follows:

a_{0k} has no meaning because the departure from the distribution center has no arrival,

$a_{jk} = \sum_{i=0}^{n-1} x_{ijk} (d_{ik} + t_{ij}), \forall j \in O, \forall k \in K$ (arrival at order delivery nodes), and arrival at route end distribution center is slightly more complicated as follows:

$a_{n+1,k} = \sum_{i \in O} x_{i,n+1,k} (d_{ik} + t_{i,n+1}), \forall k \in K$, unless

$\sum_{i \in N} \sum_{j \in N} x_{ijk} (t_{ij} + u_{ij}) > L_2$ or $\sum_{i \in N} \sum_{j \in N} x_{ijk} (t_{ij}) > L_1$, then

$a_{n+1,k} = L_3 + \sum_{i \in O} x_{i,n+1,k} (d_{ik} + t_{i,n+1}), \forall k$ which meets the preceding condition

L_1 = DOT limit on drive time before d_3 break, currently $11 * 60$ minutes

L_2 = DOT limit on workday length time before d_3 break, currently $14 * 60$ minutes

L_3 = DOT mandatory break time amount, currently $10 * 60$ minutes

Objective:

(1) Objective function

$$\text{Minimize } \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijk}$$

Subject to the following constraints:

$$\sum_{k \in K} \sum_{j \in N} x_{ijk} = 1, \forall i \in O$$

(2) Each delivery made once.

$$\sum_{j \in N} x_{0jk} = 1, \forall k \in K$$

(3) All orders start from distribution center

$$\sum_{j \in N} x_{0,n+1,k} = 1, \forall k \in K$$

(4) All orders return to distribution center

$$\sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{hjk} = 0, \forall h \in O, \forall k \in K$$

(5) All orders have one entrance segment and an exit segment, called a "flow" constraint.

$$\sum_{i \in O} \sum_{j \in N} v_i x_{ijk} \leq q_k, \forall k \in K$$

(6) Sum of all order volumes in a vehicle is less than or equal to vehicle capacity.

Note: Alternative presentation has it as follows, which I believe to be less terse, and therefore less desirable, but it is also presented this way in the literature. Because the earlier constraint requires that only one arc be active, the above constraint is equivalent to the following:

$$\sum_{i \in O} v_i \sum_{j \in N} x_{ijk} \leq q_k, \forall k \in K$$

$$s_{ik} + u_{ik} + t_{ij} - B(1 - x_{ijk}) \leq s_{jk}, \forall i \in N, \forall j \in N, \forall k \in K$$

B is any large constant that will exceed the max value of any $s+u+t$.

$$o_i \leq a_{ik} \leq z_i, \forall i \in O, \forall k \in K$$

$$o_i \leq d_{ik} \leq z_i, \forall i \in O, \forall k \in K$$

$$\sum_{i=0}^n \sum_{j=0}^n t_{ij} x_{ijk} \leq L_1 \quad \forall k \in K$$

$$\sum_{i=0}^n \sum_{j=0}^n x_{ijk} (t_{ij} + u_{ij}) \leq L_2 \quad \forall k \in K$$

(7) Arrival time of next stop is always greater than or equal to arrival time of previous stop plus the travel and unload times associated with the stop.

(8) Arrival time of every order must be within operating hours

(9) Departure time of every order must be within operating hours.

(10) The sum of all driving time up to the last stop must be within the limit for max DOT drive time.

(11) The sum of all driving time and unload time through the last stop must be within the limit for max DOT workday.

Notes

(1) This formulation of the problem is a pure representation of the problem rather than one as implemented. In the implemented model, the sum includes the minimization of calculated penalties.

(2) x_{ijk} represents an order location and not a physical location which may be shared by different orders. Identical physical location is represented as a zero distance in the arc c_{ij} .

(7) For an active arc, the constant is disabled, and so the only way for the constraint to be met is if the second arrival is greater than the first plus the additional time requirements.

(8)(9) There are no restrictions on departure and arrival time for the distribution center.

(10) DOT regulations require that driving be interrupted with a 10 hour rest period after 11 hours of driving within a 14 hour workday. Logically it would follow that one could drive for 11 hours, take a rest, and drive for an additional period (up to 2.5 hours, depending upon the unload time at the destination), but Company requirements state that all orders must be delivered within the same workday, and no delivery locations are open at 21:00. Therefore, the foregoing requirements combine to create the requirement that the last delivery location must be within 11 hours of drive time from departure of the distribution center. Note that this requirement does not state that return to the distribution center must be completed within 11 hours of drive time, only arrival at the last delivery location. Theoretically, if the vehicle arrived at the last location after an 11 hour drive, the driver could unload the order and then rest at the location for 10 hours before returning to the distribution center.

(11) The reasons for this rule are analogous to the immediately preceding constraint. DOT regulations require that driving be interrupted with a 10 hour rest period upon a driver reaching a 14 hours of work. Consequently, as with the previous constraint, a truck must be fully unloaded within the fourteen hour workday or the driver will be unable to meet the Company's requirement that all orders be delivered on the same day. Note that this constraint does not require that the truck be returned to the distribution center within 14 hours. Instead, after delivering the last order, a driver may be on his return trip when he reaches the 14 hour constraint, thereby forcing the wait period and an early morning return to the distribution center.

Software Application – The “GS490 Scheduler”

A logistics scheduler program (“GS490”) was prototyped in Python to handle the routing needs of the Company’s trucks and implement the forgoing objective function and constraints via a genetic algorithm and fitness function. The genetic algorithm is the means by which various routing paths for the trucks given the day’s orders are identified which are then evaluated with the fitness function. The pseudo code for both may be found in Appendix A, and details of the mutation/crossover functions are set forth in Appendix B. Specific important features are worth highlighting:

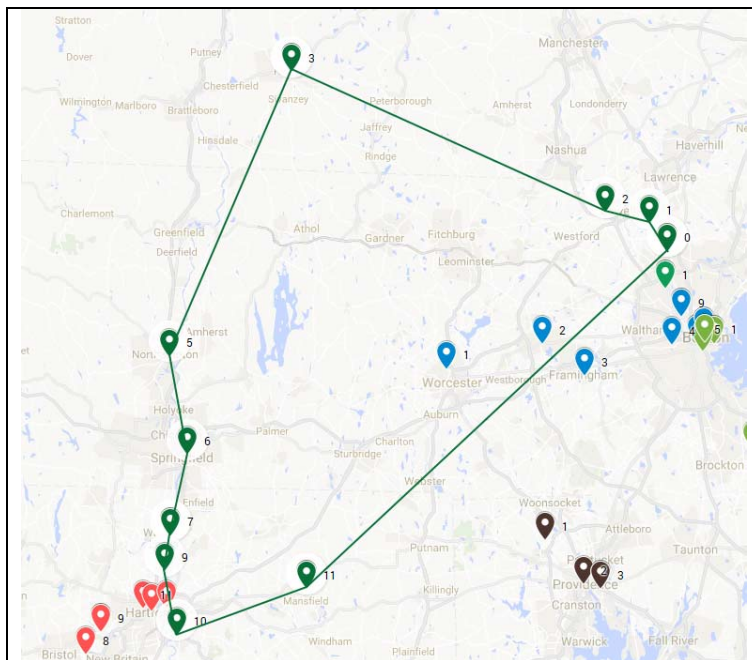
- The system is modular and allows for modification and extension as additional optimization mutations are identified. For example, there may be mutation operations which would leverage stop sequences that are close/far or dense/disparate. In the future, the software might have, for example, Company-specific optimizations that related to repeated routes or order sizes.
- The system allows for easy selection of truck count and schedule day.
- The fitness function was isolated to allow different fitness functions to be tested and used.
- Many options have been created to permit tuning of the optimization to specific needs. For example, one may tune the fitness function to avoid overnight routes resulting either from exceeding the 14 hour workday or the 11 hour rule. In addition, one may run what-if scenarios to see what performance might be obtained by adding an additional truck.
- The system automatically seeks to reduce mileage by grouping related orders (i.e., orders going to the same destination on the same day).
- The system automatically seeks to sequence spatially related destinations together.
- The system automatically initializes the system by placing large orders in a separate vehicle.
- The system allows for benchmarking; one may select a small population so that it runs quickly and provides a quick back-of-the-napkin estimate to test a particular truck count for feasibility. One may also increase the parameters for a more thorough optimization.
- The system permits easy storage of best results obtained and retrieval of those results. This allows one to start a run and stop it, and then resume the run with, at least, the best results obtained to date.
- The fitness function evaluates every route proposed by the genetic algorithm and implements all the constraints identified.

Detailed Examination of a System-Generated Truck Route

The table below shows one sample routing table for a Wednesday route (in the map above, the purple pushpins) created by the system. The map below is the plotting of this specific route. (A complete set of routes of the most optimized plans are set forth in Appendix B.)

Wednesday Truck 3

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles
		DC/Wilmington	MA		7:51:00		
1	135	Tewksbury	MA	8:00:00	8:22:00	101	6
2	138	Lowell	MA	8:35:30	9:05:30	159	9
3	225	Keene	NH	10:41:30	11:11:30	129	64
4	13	Keene	NH	11:11:30	11:11:30	117	0
5	231	Northampton	MA	12:43:00	13:13:00	151	61
6	81	Springfield	MA	13:41:30	14:11:30	140	19
7	55	Windsor Locks	CT	14:34:00	15:04:00	112	15
8	220	Windsor Locks	CT	15:04:00	15:04:00	98	0
9	253	Windsor	CT	15:11:30	15:41:30	94	5
10	167	Glastonbury	CT	16:07:00	16:37:00	140	17
11	106	Storrs Mansfield	CT	17:10:00	17:40:00	186	22
12	193	Storrs Mansfield	CT	17:40:00	17:40:00	118	0
13		DC/Wilmington	MA	20:01:00			94
Totals						1545	312



Plotting the above table spatially, one can see visually the schedule is highly optimized. Sequence numbers are identified next to the push pins. The system identified that the Keene, NH order (stop 3) was best placed on the green route because of its proximity to pins 4 (not visible as it is a multiple order at the same location) and 5 at Windsor Lock, Ct. The zero push pin is the start/end distribution center. Note: Lines show sequencing, not actual road travel.

For interactive examination, visit:
<https://drive.google.com/open?id=1ZIG9GC2uQ1YgAaEkzcNTTcVQajY&usp=sharing> (remove any spaces)

The table shows that Company requirements were properly implemented by the GS 490 system:

- Routing began so that the truck arrived at its first destination at 8:00am.
- All unloading at the last stop was completed before 18:00; in this case, the last stop was completed at 17:40.
- The driver had less than a fourteen hour work day. In this case, the driver's day was twelve hours and ten minutes (7:51:00 to 20:01:00).
- The driver did not drive more than ten hours within the workday. In this case, the driver had slightly more less than eight hours of driving the 312 miles.
- A minimum of thirty minutes of unload time was required for each stop.
- Aggregate volume for the truck did not exceed 3200 ft³.

In terms of optimization, the following items should be noted:

- The route sequences adjacent locations and optimized the Keene, NH order.
- Multiple orders to the same locations (e.g., orders 225/13, 55/ 220, 106/193) were automatically grouped together sequentially to minimize distance and unload times.

Prototype System

The GS490 Genetic Scheduler was created to create a complete schedule for a given day for which the one route above is just a part. Developing the prototype allowed the technical department to determine what process and technologies were required to create optimized routing schedules. Specifically, the department took the following steps:

- 1) Queried and extracted data from the data warehouse to create average orders to each center for a representative week.
- 2) Transformed the data to relate the necessary orders to distances, calculate distances between cities for daily orders, and enhance the data with pre-calculated values.
- 3) Created suitable load files for loading into a scheduling program.
- 4) Coded a genetic algorithm in the Python language to read the load files and optimize a daily schedule.
- 5) Exported the schedule and checked it for accuracy.


Data Preparation

Two data tables involving basic order information and city-to-city distance information were extracted from the Company's data warehouse. This data was then pulled into an MSAccess database for ease of manipulation outside of the data warehouse. Using additional queries and generation of additional metadata, two data tables were created for use by the logistics' algorithm with the following schemas:

Order Data

Field Name	Data Type	
orderid	Number	OrderID
fromzip	Text	DC zip
fromzipid	Number	DC location ID
tozip	Text	Order Location zip
tozipid	Number	Order location location id
tolon	Number	lon for mapping
tolat	Number	lat for mapping
tocity	Text	Order location city
tostate	Text	Order location state
cube	Number	Volume in cubic feet
dayofweek	Number	Delivery day of week
strequired	Number	Whether straight truck required
milesfromdc	Number	Delivery location distance from DC
minutesfromdc	Number	Delivery location time from DC
minutesunloadraw	Number	Raw time required to unload delivery
minutesunload	Number	Actual time required to unload
multipleorder	Number	Multiple order to same location
tripleorder	Number	Triple order to same location
closest1	Number	Closest location id with delivery same day
closest2	Number	Next closest location, delivery same day
closest3	Number	Third closest location, delivery same day
closest4	Number	Fourth closest location, delivery same day

Distance Data

Field Name	Data Type	
ID	AutoNumber	record id
fromid	Number	Location id from
toid	Number	Location id to
distance	Number	distance
rank	Number	the destination's rank order of distance
dayofweek	Number	the day of week for this pair of distances
		

The primary enhancement to the order data is the full detail of location information as well as pre-calculated distance and travel minutes from the distribution center. In addition, unload time has been pre-calculated. Based on the daily order data, orders were flagged if they were part of a double or triple package set of orders to the same destination on the same day. Finally, the four closest store locations adjacent to this delivery location *for deliveries occurring on the same day* were identified and added into rank-ordered columns in the data table. With this refined data table, the logistics scheduler could more easily and more quickly test proximity-related optimizations.

Secondly, the original data warehouse distance table contained distances from every delivery location to every other delivery location. While complete, this yields a data table of many thousands of records, most of which are unused because the stores receiving orders on any particular day range from a low of forty three (43) to sixty three (63) stores, out of a possible one hundred twenty three (123) stores. Consequently, a new distance table data was output with only the location-to-location distances relevant for each specific day of week based on the orders. Pre-processing the data in this manner reduced the total size of the file by fifty percent (50%), and because only items for a particular weekday would be searched, the data format reduces the computer search time of the logistics algorithm.

Additional Improvements to Exploration of Possible Routes

Many genetic algorithms get “stuck” on mediocre solutions and are unable to locate the best route. Considerable effort was undertaken to ameliorate these limitations, and a number of heuristic improvements were made to the base genetic algorithm: 1) High-Volume Order Initialization; 2) Multiple Order Mutations; 3) Adjacent Order Mutation; 4) Equal Child Rule; 5) Simulated Annealing Inspired Offspring; and 6) Parallel Individuals/Proportional Breeding.

1) High-Volume Order Initialization

Once the system was running, early experiments showed that large packages posed a large problem for optimization. In short, the system had to isolate large packages that caused truck volume overages, and the use of random mutations to isolate the large packages could take tens of thousands of generations. Simple analysis showed that it would be much, much easier to fill up a truck containing a single large package that it would be, with random assignment, to reorganize tens of packages simultaneously to achieve the same goal.

Thus, a high-volume initialization procedure was established. After random assignment of the packages, the system queries the day’s data for high-volume orders that are greater than or equal to 2400ft³. If such orders are found, then all other packages in such vehicle are relocated randomly to other trucks, excluding any trucks that had previously been found to contain high-volume orders. This changed the distribution of the packages from an equal distribution to a skewed one, but with very favorable results. Instead of taking tens of thousands of iterations to organize the volumes into a conforming state, it would take only a few generations, and this sped the optimization process.

2) Multiple Order Mutations

Analysis of the data showed that on any given day, multiple orders had to be delivered to the same location. If such orders were placed sequentially in the same truck, then not only would travel time be reduced, but packages could be consolidated for purposes of minimizing unload times. Consequently, mutations for two and three-order multiples were developed. The use of this mutation boosted the speed of optimization significantly and the results are clearly visible in the order sequencing seen above.

Some thought was given to initializing all vehicles in this manner, but some consideration imagined scenarios where the mileage benefit or the volume benefit, or both, would not be realized from

consolidation on the same vehicle. For example, one might have three orders that are each larger than 1000 cubic feet (thus exceeding the minimum unload times) that are delivered on a major departure road from the distribution center. In such a case, there are likely to be no costs to putting the orders on multiple trucks as opposed to one truck. Consequently the design implements a mutation functions so that this sequencing may occur in a random manner over the course of the optimizing process.

3) Nearby Order Mutations

The nearest-neighbor heuristic is an acclaimed heuristic for traveling salesman problems. The idea is a simple one; if you want to minimize your travel, go to the next closest location. This simple heuristic led to the development of a probabilistically relaxed version of the nearest neighbor heuristic referred to as the nearby-order mutation. This mutation applies to a randomly selected order, and the system will move either the closest, next-closest, third closest, or fourth closest order (determined by a probability) adjacent to the selected order.

The reasoning for a probabilistic selection rather than the hard-and-fast closest selection was a simple one. A rigid selection of always choosing the closest neighbor could lead to recurrent trapping in a local minima situation. Instead, the system explores other possibilities that may (or may not) yield higher fitness values. Even if the selection of the third closest neighbor is less than optimal, continued iterations will eventually experiment with the closer neighbor, and, if more optimal, it will be selected. Learning comes more slowly, but at the benefits of learning more thoroughly by not rigidly excluding other choices.

4) Equal Child Rule

In some genetic algorithms tournaments are used to determine which individuals comprise the next generation and offspring will replace parents only if they are more fit than the parents against whom they are tested. The “only the best survive” continually improves until it doesn’t, limited by its purging of individuals who may not be more fit at the moment but which may harbor a fortunate insight into the data. Because the offspring are the mechanisms for exploring the search space to find better routing opportunities, it is important to encourage the propagation of wide and varying offspring while leveraging their existing structural strengths.

Consequently, for this project, a simple rule was implemented. If a child has the same fitness value as (or better than) a parent, then the child replaces the parent. This is based on a simple insight that it is unknown whether the parent or the child has a more fortunate sequencing that will lead to the most optimal state. Although little is known about the child, it is known with certainty that the parent has failed to produce a more-fit offspring (because if it had, it would already have been replaced). Thus, the balance falls to the child, and equally good children routes always replace parent routes.

This simple heuristic principle results in considerable movement across the search space as edges are swapped and genes mutate continuously even if they do not yield an obvious benefit at the time. As this continuous mutation pattern continues, eventually a more fit candidate is found and selected, and the fitness value ratchets downward. This has shown experimentally to result in a more even hill-

descent across all candidates, and often multiple candidates will be within a few points of one another at any given generation.

5) Simulated Annealing Inspired Offspring

Simulated annealing algorithms are reportedly more resistant to local minima because they have a mechanism to accept worse solutions; as a result, they systematically enable exploration of the search space as well as exploitation of the current set of sequencing. It does this through a probabilistic adjustment to the scores accepted.

A less probabilistic, but similar, approach was followed in the GS490 Scheduler. While downward movement occurs, exploitation of the current route structure is pursued by rejecting inferior offspring and accepting only better or equal offspring. However, if the system fails to find an equal or better offspring within a specified number of generations, then the algorithm accepts the possibility of accepting a worse offspring. The more iterations that pass, the less fit the offspring can be and still be accepted. Thus, when repeated mutations fail to produce an equal or better offspring, the algorithm self-adjusts and selects a worse offspring to replace the parent. The idea is that a somewhat-inferior offspring may lead out of the local minimum and doing so is better than getting stuck. The less-fit offspring may have substantial mutations which generate (temporarily) a lower fitness value but open up new possibilities for mutation.

A true simulated annealing algorithm uses a probabilistic model. This system increases the acceptance of an inferior offspring in a linear relationship to the time that the last offspring occurred. Experimental results show that even this simple algorithmic adjustment appears to improve overall fitness values. Implementing a true simulated annealing algorithm in this context would be an opportunity for improvement of the system.

6) Parallel Individuals/Proportional Breeding

The initial design of the GS490 used proportional breeding relative to overall fitness. Individuals were selected as candidate children based on their proportional fitness to the entire population. Experience showed that with small populations, random chance could shift the population to a single route pattern (or genetic line). If a few individuals had some early fortunate mutations, they could rapidly multiply and drown out other lines.

Proportional breeding combined with the genetic algorithm's tendency to get stuck in local optima pushes one to use more sizable populations (e.g., 200 individuals) because larger populations have greater diversity and are more resistant to the foregoing problem. Because the evolutionary process can be a slow one given the vagaries of the data which is processed, a run of 25,000 generations might be required to obtain a reasonably optimized solution. Thus, large population counts combined with the sizable iteration counts mean that a good result can require hours of computation.

Consequently, the GS490 allows for either proportional breeding or a concept of parallel solutions. This is akin to island populations that survive even though alternative competitors have higher fitness values.

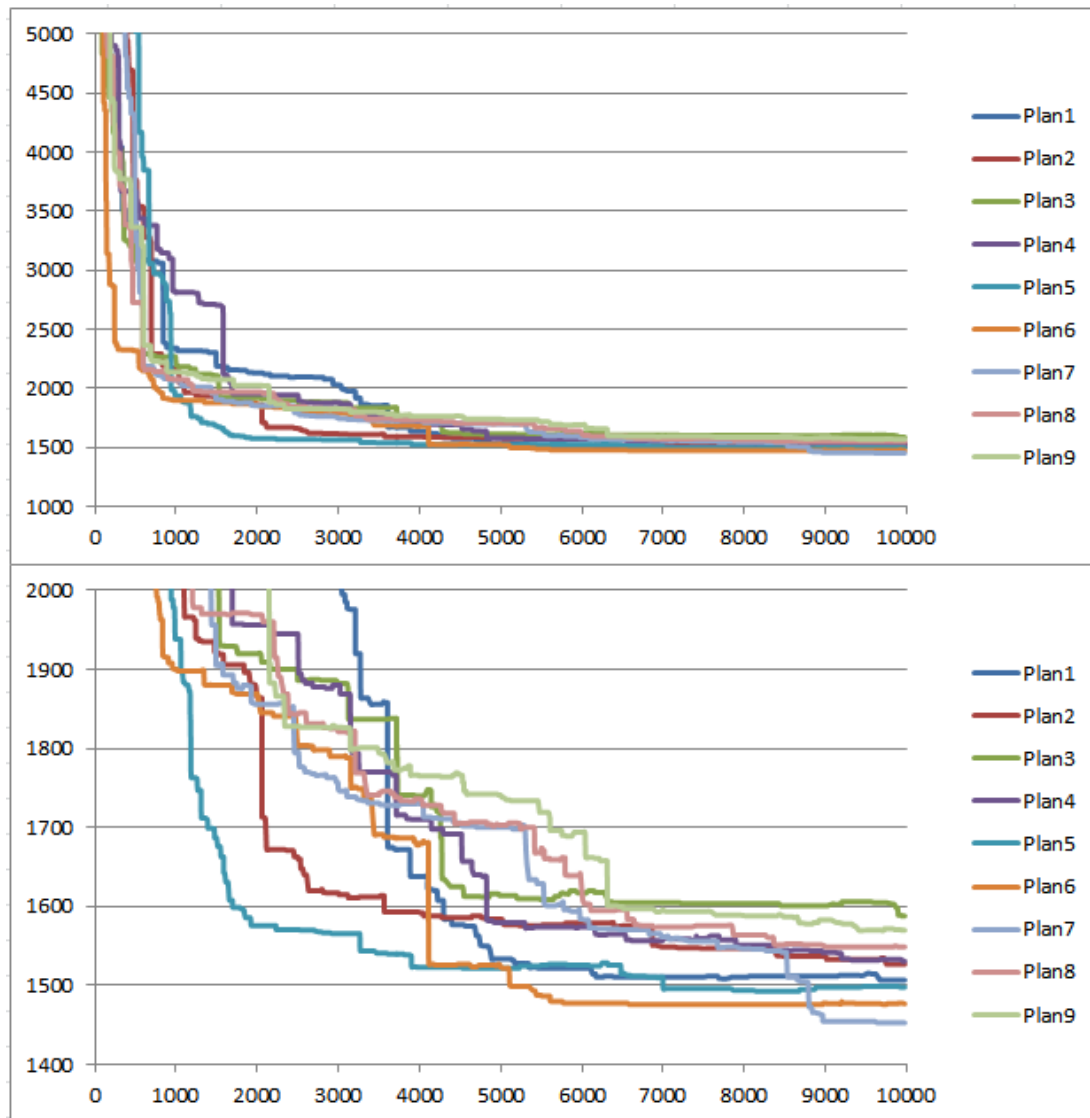
Experimental results with the prototype system show that very fit individuals' progress may slow down and get stuck in a local optimum for many generations; in the meantime, a worse-fit cousin with a completely different route pattern will benefit from a fortunate mutation arising from the Equal Child Rule. Consequently, a competitor who was substantially less fit for 5,000 generations may quickly descend and become the most fit. By having independent routes which mutate and evolve independently, much smaller populations can achieve a high level of performance.

Within the parallel solution context, a "Captain's Share" rule was adopted. (On pirate boats in the 19th century, the looted bounty was reportedly divided into $n+1$ portions for n pirates, one share for each pirate plus two for the Captain.) When one chooses the parallel solution process in the GS490 software, the most-fit individual is given an extra clone. Thus, in a population of 10 individuals, 9 will be independent parallel solutions, while the 10th will be a clone of the most-fit individual. To the extent that the clone improves, then that improvement gets copied back to the primary individual. This gives the most-fit individual plan a double opportunity to improve. This heuristic results in a faster descent of the most-fit individual in the first few thousand generations, thus allowing one to ballpark whether a particular vehicle solution is possible, and one may do so with a population as small as three. Longer runs show that the lead position changes frequently within the first 5,000 generations, and may change continually as the generations pass, and this is expectable if the search space is expansively examined.

(This portion left intentionally blank to avoid a break on the next page.)

Assessment of Heuristics Performance

Because this problem is NP-hard, it can be difficult to know the extent to which the search space is fully examined. One way to assess the effectiveness of the heuristics is to examine the algorithm's consistency in its production of a solution. Smaller variance with continued decline among runs may suggest resilience to getting stuck in local minima. The following graphs explore this idea by plotting nine independent route plan scores as they descend across 10,000 generations.



As you can see, within 4,000 generations, 100% of the plans have scores under 1800 miles, and by 6,000 generations, all are less than 1700 combined miles. While the plans optimize at different rates, none appear to get stuck until they achieve scores of 1600 or less. Even then, they may not be stuck at all as the best plan did not optimize until about 9,000 generations. The worse plan was within 10% of the best plan, and 2/3 of the plans were within 5% of the best plan. Collecting statistics like these allows one to evaluate proposed improvements to see if the proposals are furthering the expansion of the search space, fostering descent, and narrowing the variance between the plans.

The Fitness Function

The system's Fitness Function is the plan evaluator and, through a system of penalties, it implements all the operational constraints. It examines a plan like the one above and uses all the information about the distances, stops, and orders to calculate mileage, travel time, unload time, aggregate container volume, and operational compatibility (e.g., if the order can be unloaded by 6:00pm on the same day), and overnight routes. A valid plan meeting all criteria will simply be the total distance in miles, and because the system promotes plans with lower scores, the total distance of the plans is driven downwards towards the most optimal value.

For invalid plans, the fitness function will start with a total mileage score as the base score and will add penalties for any of the following: volume overage, invalidity due to DOT violations, and overnight stays. Because the score is a combination of mileage plus penalties, a good plan with high penalties can be considered worse than a plan with higher miles but no penalties. Because the system prefers lower score values, the invalid plans disappear from the routing plans as the scores descend. The penalties are discussed below.

Volume Overage Penalty: $100,000 \text{ miles} + 2 * \text{overage in ft}^3$

Reason: Originally, the base penalty was set to a small value (e.g., 100). Experiments demonstrated quickly that the genetic algorithm could get trapped in an invalid plan where the overage was less than ten cubic feet. Because of the combination of package sizes and locations, the cost of moving any of the packages was more than one hundred miles, and so the vehicle plan could not determine a way out of this excessive volume situation. As an essential plan criterion, the volume violation had to be eliminated, and consequently, a high penalty value was assigned. Once a pattern has removed an overage condition, the system will not reinstate it regardless of the mileage benefit. This penalty, combined with the initial high-volume order initialization quickly resolves all volume overage issues.

Invalid Route due to Eleven Hour Violation Penalty: 10,000

A very important penalty arises from need to prevent invalid plans due to an eleven hour drive-time violation. The Company requires that all packages be delivered the same day, and the DOT requires that a break of ten hours follows a drive time of eleven hours within a fourteen hour workday. The intersection of these two rules means that when eleven hours of drive time occurs before the arrival at the last location, an invalid plan has been created. The penalty assigned to this is a 10,000 mile penalty, which means that an invalid plan of this sort is usually remedied within the first 100 generations and will not be reinstated. There is concern, however, that the penalty amount may be too high, which is resulting in some areas of the search space receiving less exploration that might be considered with a lesser value. Optimization of this value is an area of opportunity for future work.

Workday Too Long Penalty: $500 + 2 * \text{OverageInMinutes}$

A less troublesome invalid plan occurs when the last package has not been unloaded by 18:00 as required by the Company. A plan with this violation is less troublesome because moving some orders from the invalid route to another route can be accomplished very easily and at low cost in mileage. There are many factors which affect the number of minutes in a driver's work day. Specifically, the miles driven generate minutes, and each order has an unload time, and multiple orders can have zero

unload times. The variety of ways that the algorithm can optimize the workday minutes means that the penalty may be kept relatively low, and experimental results show that the value of 500 is sufficient for inhibiting invalid plans.

Initially, the value was set to one hundred (100), but there were concerns that the system might optimize a plan in such a way that it could not find a way out of an invalid plan. For example, if a particular route was created such that no set of mutations could create an alternative plan within only a 100 mile increase, then the system would be stuck with an invalid plan. There may be some benefit to further optimization of this penalty value.

Overnight Drives Penalty: $50+2*\text{OvernightMinutes}$ (Optional)

The Overnight Rule and the Fourteen Hour Workday Rule may require a driver to take the DOT-mandated ten hour break during the drive back to the distribution center. The costs associated with overnight drives are not fully understood at this time. Overnight drives may be problematic: 1) the trucks are not safely docked at the distribution for maintenance as well as preparation for the following day; 2) overnight drives keep the driver away from family and put a strain by sleeping in the cab of the truck; 3) labor agreements may prohibit such driving or have enhanced compensation for it; 4) overnight cabs are special equipment that may be in short supply during high-transit times of the year; and 5) federal regulators may engage in additional oversight to ensure compliance with overnight driving for companies that engage in it extensively. Consequently, the system contains an optional penalty. When enabled, this shifts routes to non-overnight routes when the routes are only a few extra miles. For example, if an overnight route only requires 10 extra minutes on the road after the break, then a route up to 70 miles longer would be preferred. The penalty is adjustable to allow for fine-tuning based on Company needs and policy.

The Mutation Process

In the early development of the software, the decision was made to use asexual reproduction as the method of exploration of the solution space. Asexual reproduction occurs naturally within a number of animal species living today, and so it serves evolutionary purposes even though sexual reproduction may be more commonplace. By making children plans from mutated versions of a single adult plan, a number of specific benefits were obtained: 1) simplified crossover and mutation processes; 2) robust exploration of the mutation operations that might yield better search performance; 3) smaller populations of independent parallel plans; and 4) the structuring of mutation process in new ways.

By keeping all crossover/mutation operations within a single genome, no management of the orders was required. Orders and sequences of orders could be moved around within the set of routes knowing that an invalid plan could be created. Where plans are created from two plans, explicit order management is required to prevent a plan's duplication or elimination, and the journal literature is filled with discussion of ways to resolve conflicts. Instead of spending the time coding and debugging these more complicated operations which might or might not yield much benefit over asexual reproduction,

the time was spent analyzing the problem for heuristics (e.g., large truck initialization) that would generate large value.

With asexual reproduction, the focus became the creation of robust crossover operations within the single individual in order to adequately examine the search space. Consequently a substantial variety of edge swaps and moves were developed to simulate the benefits of sexual genetic crossover. Finally, individual gene moves and swaps simulate random mutation while preserving all orders in a sequence.

Because individuals were reproducing asexually, smaller populations were possible because diversity could be maintained with even a small independent set of individuals. This allowed easy experiments with new edge swaps and mutations.

Finally, an unanticipated benefit of this approach was the development of new ways of thinking about the mutation strategy. Specifically, different kinds of stresses could be placed on the route patterns at different times. Within the natural biological context, animal life is subject to a variety of pressures, including stresses from food, shelter, predators, injury, and disease. A fit individual may survive any of these particular stresses. For example, if a new predator is introduced, a single mutation may offer an extra capability that allows the individual to have offspring before being eliminated; this can give rise to various “arms races” between species (see: https://www.washingtonpost.com/news/speaking-of-science/wp/2016/06/10/how-an-evolutionary-arms-race-with-snakes-turned-newts-super-toxic/?utm_term=.ac72c0f86526). An animal whose genome subjects it to multiple stresses at the same time (e.g., unfortunate genes that leading to nutritional stress, combined with weakness vis-à-vis a predator), will probably result in a bad outcome for the individual; it is simply less fit and selection of the fittest will weed it out.

In implementing the genetic algorithm of the software, some consideration was given to how the various edge swaps and moves and individual gene mutations and moves emulated these sorts of distinct stresses. An edge move from a middle of one vehicle to a middle of another vehicle is a very different kind of genetic mutation than a beginning to end (with inversion) edge swap or an end-edge to end-edge swap. The examination of mutation patterns raises the question of whether certain kinds of swaps are more beneficial during specific evolutionary phases of the routing patterns. Experimentally, it became obvious that a lot of mutations combined with lots of edge swaps could generate a lot of search exploration, but at the expense of creating substantially inferior child plans.

An alternative approach was to have a variety of mutational patterns implemented with randomness. Some mutational patterns might emphasize equal weighting of individual gene mutations and edge moves. Other patterns might focus on only a few mutations and edge swaps, while others might have only edge swaps without mutations. This appears to produce a reasonably robust result in terms of avoiding local minima. The rigorous examination of this heuristic might give rise to some interesting research in the field.

Conclusion

The department is pleased with the results of the GS490 Genetic Scheduler and believes that its results may serve the company as a valuable analytical tool in the following ways:

- Assessment of proposals and review of company costs
- Monitoring of third party common carrier performance
- Assessment of costs for additional store locations or distribution centers
- Assessment of schedule/cost impact from changes in store hours
- Assessment of schedule/cost impact from changes in truck equipment
- Assessment of schedule/cost impact from planned road maintenance/changes
- Labor discussions with transportation unions by providing details of economic impact
- Capital acquisition decisions
- Corporate cost planning

Potential opportunities for further improvement of the GS490 software and its algorithm would include the following areas:

- Full implementation of a simulated annealing processing layer
- Collection and analysis of transformation data to categorize mutations as to whether they serve exploitation or search expansion functions
- Analysis of to determine when exploitation or search functions should be used
- Expansion of the mutation profiles based on exploitation search functions
- Optimizing probabilities of nearest neighbor mutation through simulation testing
- Optimizing volume size settings for large order initialization process to increase speed to solution
- Additional rare mutations.
 - Inward-out sequence transformation, e.g., 123456 becomes 321654
 - Other edge-related transformations, e.g., 1234 becomes 3142
 - Middle to edge moves, i.e., taking the middle of one truck route and placing it at the beginning or end of another truck sequence
 - Middle to edge swaps; as above, but with a swap instead of a move
- Addition of heuristics for dealing with larger truck counts
- Addition of classical sexual reproduction within the genetic algorithm with modern heuristics of how to combine the differing patterns. Some thought has been given to maximizing genetic diversity through the use of group sex rather than dual sex. Once one has the order management code written, one could draw children plans from the collective set of plans.

Appendix A

GS490 Pseudocode

General Algorithm

- Read in all data files and initialize system
- Create the initial population of random routes
- Reallocate routes to give high-volume (i.e., >2400 ft³) packages an entire truck (the system will then optimize the reallocation of packages in other trucks to fill the truck).
- Create the breeding population if proportional breeding enabled
- Loop for required generations or until fitness criteria achieved
 - Create a child population of the same size as the current population:
 - Always clone the most fit member of the parent population and make that the last child in the child population.
 - If proportional breeding, randomly select parents based on proportional fitness.
 - If parallel individual breeding, then each parent acts as the base for one child.
 - Apply the mutation rules to each child in the population
 - Create a unique mutation profile from the various mutation functions in Appendix A for the individual
 - Apply the mutations
 - Calculate the fitness of the individual
 - Determine if the parent plan should be replaced with the new individual's plan
 - If the child plan is more fit than the parent plan:
 - Substitute the child plan as the new plan
 - Reset the spawn counter
 - If the new plan is the top plan, save it as the top item for printing purposes
 - If search expansion is enabled:
 - If the child plan = parent plan:
 - Substitute the child plan as the new plan
 - Reset the spawn counter
 - Else if the spawn counter > specified threshold:
 - If the difference between the two plans is less than the linear amount specified by the spawn counter:
 - Substitute the child plan as the new plan
 - Reset the spawn counter
 - Calculate the proportional fitness of each individual in the population and create the breeding population if proportional fitness is used
 - Output any diagnostic information about the generation to the console
- Output the top item best plan information

Fitness Function

- Initialize penalty values for excess volume, out-of-hours, and overnight stays
- Set total fitness value to 0
- Set total distance to 0
- For all trucks in the master plan:
 - For each stop in a truck's plan:
 - Get the details associated with the order associated with the stop
 - If the first stop
 - Store the distance for this stop equal to the distance of this stop from the distribution center
 - Store the volume for this order
 - Store the first travel minutes equal to the minutes traveled to the first stop
 - Else not the first stop:
 - Search for the distance between this stop and the previous stop
 - Store the distance measurement, the travel time, and the volume required for this order
 - Calculate and store the unload minutes
 - If the new location is different from the previous location:
 - Calculate unload minutes with 30 minute minimum rule
 - Else same location:
 - Calculate any overage to the 30 minute minimum
 - Save the current location as the previous location.
 - With the truck route complete, calculate the following sums:
 - Calculate total unload minutes as the sum of all truck unload minutes
 - Calculate total volume as the sum of all truck volume amounts
 - Calculate the "middle" travel minutes with the formula:
 - $\text{MiddleTravelMin} = \text{TotalMinutes} - \text{FirstStopMinutes} - \text{LastStopMinutes}$
 - Calculate $\text{TotalMiddleMinutes} = \text{MiddleTravelMin} + \text{SumUnloadMinutes}$
 - Calculate $\text{TotalRouteMin} = \text{TotalTravelMin} + \text{SumUnloadMinutes}$
 - If $\text{FirstTravelMinutes} > 240$ (because you can't travel work more than 14 hours, and the work day has 10 core hours):
 - Add the excess above 240 minutes for the first stop to the $\text{TotalMiddleMinutes}$
 - Set the first travel minutes to 240
 - If the $\text{TotalMiddleMinutes} > 600$ (10 hour work day):
 - Calculate the $\text{ExcessMinutes Penalty}$
 - Else
 - Set the $\text{ExcessMinutes Penalty}$ to 0
 - If $\text{TotalVolume} > 3200$
 - Calculate the $\text{ExcessVolumePenalty}$
 - Else

- Set ExcessVolumePenalty to 0
- Set OvernightMinutePenalty to 0
- Calculate OvernightMinutes = TotalRouteMin – 14*60 (14 hour work day max)
- If OvernightMinutes > 0
 - Set daytimetravelmin = sum of all travel minutes in route – overnight minutes
- Else
 - Set daytimetravelmin = sum of all travel minutes in route
- If daytime travel minutes < 11 hours
 - Set initial fitness value to sum of all travel minutes
- Else
 - If we want ALWAYS to forbid any plan that exceeds the 11 hour rule (the alternative is to use a 10 hour break where appropriate)
 - Set route fitness value to sum of all travel minutes + Large Penalty
 - Else
 - Calculate the sum of all travel minutes except the final trip to DC
 - If travel minutes except last DC trip > 11 hours (which means that it is an impossible route)
 - Set route fitness value to sum of all travel minutes + Large Penalty
 - Else (we simply have a overage in the trip back)
 - Set overage minutes = travel minutes – 660 (11 hours)
 - If overage minutes > previously calculated overnight minutes due to the 14 hour rule
 - Set overnight minutes = overage minutes
 - Set route fitness to be sum of travel minutes
- If overnight minutes > 0 and overnight penalty option is enabled (which means we want to try to select no-overnight plans within a specified tolerance)
 - Set overnight penalty to overnight base penalty + 2*overnight minutes (the base will establish the threshold for alternatives, i.e., if the base is 25, then plans with no overnight stay that are 25 miles or less will be selected)
- Set route fitness value to the route fitness value + overnight penalty
- If VolumePenalty>0, then route fitness value = route fitness value + VolumePenalty*2 + VolumePenaltyBaseAmount
- If ExcessMinutesPenalty>0, then route fitness value = route fitness value + ExcessMinutesPenalty*2 + ExcessMinutesPenaltyBaseAmount
- Return the Sum of all Trucks' TruckRouteFitness values

APPENDIX B

Details of the GS490 Genetic Scheduler Mutations

The GS490 scheduler is a custom Python application using a genetic algorithm. The software process is called a genetic algorithm because it simulates the randomness of evolution, combined with a fitness function that acts as nature's survival of the fittest. In the GS490's software design, each truck is considered a chromosome, and each stop on the truck's route is a gene. Thus, for a five truck scenario with forty (40) stops, the system would dynamically generate five chromosomes and populate them evenly with a random assortment of stops. An example of an initial route might be as follows. Please note that the starting and ending location (the distribution center) is always implicit and therefore never managed in the chromosomal sequence.

Stop	Truck 1	Truck 2	Truck 3	Truck 4	Truck 5
1	32	33	5	25	8
2	28	21	40	31	13
3	18	15	24	4	38
4	36	39	34	22	9
5	3	14	11	2	26
6	30	29	20	1	16
7	12	6	27	35	10
8	7	17	19	23	37
9					
10					

(Example start)

These routes will then be modified through a number of different mutation functions defined as follows.

(remainder of page left blank for ease of reading)

Delete/Add

The most basic operation is to delete a stop from one truck's route and add it to another truck. For example, assume that the system randomly selected Stop 5 on Truck 1's route (location 3), and randomly selected to move it to Stop 7 on Truck 4's route (currently occupied by location 35):

Stop	Truck 1	Truck 2	Truck 3	Truck 4	Truck 5
1	32	33	5	25	8
2	28	21	40	31	13
3	18	15	24	4	38
4	36	39	34	22	9
5	3	14	11	2	26
6	30	29	20	1	16
7	12	6	27	35	10
8	7	17	19	23	37
9					
10					

Once the deletion and insertion has been made, the routes would be as follows:

Stop	Truck 1	Truck 2	Truck 3	Truck 4	Truck 5
1	32	33	5	25	8
2	28	21	40	31	13
3	18	15	24	4	38
4	36	39	34	22	9
5	30	14	11	2	26
6	12	29	20	1	16
7	7	6	27	3	10
8		17	19	35	37
9				23	
10					

(remainder of page left blank for ease of reading)

Same Truck Swap

The next basic operation is a swap within the same truck's route. For example, using the previous route plan in the preceding example, suppose that the system randomly selected Truck 2, Stops 2 (location 21) and 6 (location 29). After the swap, the routes would be as follows:

Stop	Truck 1	Truck 2	Truck 3	Truck 4	Truck 5
1	32	33	5	25	8
2	28	29	40	31	13
3	18	15	24	4	38
4	36	39	34	22	9
5	30	14	11	2	26
6	12	21	20	1	16
7	7	6	27	3	10
8		17	19	35	37
9				23	
10					

Different Truck Swap

A variant of the above truck swap is the different truck swap. In this case, suppose that the system randomly selected the Truck 2, Stop 6 (now occupied by location 21) with Truck 5, Stop 8 (location 37):

Stop	Truck 1	Truck 2	Truck 3	Truck 4	Truck 5
1	32	33	5	25	8
2	28	29	40	31	13
3	18	15	24	4	38
4	36	39	34	22	9
5	30	14	11	2	26
6	12	37	20	1	16
7	7	6	27	3	10
8		17	19	35	21
9				23	
10					

As you can see, with multiple operations, the stops may move around the entire master plan. Location 21 had started in Truck 2 Stop2, but is now at Truck 5 Stop 8.

One can think of these individual swaps and the delete/add as the equivalent of gene mutation in the natural world. The genetic sequence changes at a single location. However, because the Company has specific orders that must be fulfilled on a particular day, it is important to maintain the correct population of genes (i.e., stops). For this reason, the system cannot simply overwrite one stop with another but must do something with the other stop, either move it up or down or exchange it.

Asexual Crossover

As the system develops its routes, a lot of information will be stored in the sequencing of the various stops for a particular route. Simply swapping individual genes (stops) would destroy the information that had been assembled and stored. In nature, this information encoded in the genome is passed from parent to offspring via chromosomal crossover where two parents' DNA mixes in segments to form the genes of the new child. (Technically speaking, it is the grandparent's genes that are mixed, i.e., recombined, together before being halved for purposes of sexual reproduction.) Because of the technical complexity associated with combining partial truck routes of one master plan (as seen above) with a different master plan (not shown), this was avoided. Such combination would be akin to sexual reproduction in the natural world.

For this genetic algorithm, a more modest approach was taken to create offspring master plans solely from parent plan. This may be likened to asexual reproduction in the natural kingdom seen in bacteria and known as binary fission, but asexual reproduction is also seen in animals like the copperhead snake. With this approach the sequencing stored by the parent plan is preserved, though moved.

Same Truck Sequence Move

One form of this genetic crossover in the system is the swapping of a sequence within the same truck's route. For example, assume that the system chose to move Truck 5 Stops 5-7 (Locations 26-16-10) in front of Stop 1 (Location 8) from the plan in the preceding section. The route would then look as follows:

Stop	Truck 1	Truck 2	Truck 3	Truck 4	Truck 5
1	32	33	5	25	26
2	28	29	40	31	16
3	18	15	24	4	10
4	36	39	34	22	8
5	30	14	11	2	13
6	12	37	20	1	38
7	7	6	27	3	9
8		17	19	35	21
9				23	
10					

(remainder of page left blank for ease of reading)

Different Truck Sequence Swap

In addition to swapping within the same truck route, the system can also swap between truck routes. An example would be if, from the preceding section's plan, it selected Truck 4, Stops 1-3 (sequence 25-31-4) and moved it to Truck 1, Stop 5. (Location 30). After the move, the routes would look as follows:

Stop	Truck 1	Truck 2	Truck 3	Truck 4	Truck 5
1	32	33	5	22	26
2	28	29	40	2	16
3	18	15	24	1	10
4	36	39	34	3	8
5	25	14	11	35	13
6	31	37	20	23	38
7	4	6	27		9
8	30	17	19		21
9	12				
10	7				

As you can see, the route lengths in the child plan may be quite different from the parent.

Same Truck Inverse Sequence Move

In addition to maintaining the orientation of the sequence, it is also important to allow the sequence to invert. This is because the starting and ending stops are the distribution center, and so a move from a beginning of one route that may be placed at the end of another truck's route, provided that it has been inverted. For example, assume the system randomly chose Truck 1 Stops 9-10 (sequence 12-7) from the preceding section's plan and inverted it and moved it to Truck 3 Stop 2 (Location 40). After the inversion and move, the routes would be as follows:

Stop	Truck 1	Truck 2	Truck 3	Truck 4	Truck 5
1	32	33	5	22	26
2	28	29	7	2	16
3	18	15	12	1	10
4	36	39	40	3	8
5	25	14	24	35	13
6	31	37	34	23	38
7	4	6	11		9
8	30	17	20		21
9			27		
10			19		

Double Orders/Triple Orders

Due to the variety of orders received, sometimes the Company receives more than one order destined for the same location on the same day. Placing these orders on the same truck so that they are delivered together saves miles and unload time. The system is intelligent and knows which orders for the day are double or triple orders. When this function is called, the system randomly selects from one of the multiple orders. It then locates the other related orders and moves them to the selected order's location.

For example, assume that the following pairs of orders were related: 33-1, 4-6, and 27-19, and the system randomly chose Location 33 from those pairs. It would then identify the related pairing (Location 1 at Truck 4 Stop 3) and move it randomly in front of or behind Location 33. After this function is called, the routing would look as follows:

Stop	Truck 1	Truck 2	Truck 3	Truck 4	Truck 5
1	32	1	5	22	26
2	28	33	7	2	16
3	18	29	12	3	10
4	36	15	40	35	8
5	25	39	24	23	13
6	31	14	34		38
7	4	37	11		9
8	30	6	20		21
9		17	27		
10			19		

(remainder of page left blank for ease of reading)

Optimization Function – Nearest Location

One of the most basic heuristics used in logistics routing is the idea that the closest location to a stop may be the most suitable previous or next stop. Consequently, a function was created to implement this heuristic.

As described in the report, the data tables used by the system contain the closest cities to every order. When this function is called, a stop is randomly selected. The system then selects either the closest through fourth closest location as the location to change. It locates that designated stop in the routes and removes it and places it randomly in front of or behind the initial location chosen.

For example, assume that Truck 4, Stop 4 was chosen (Location 35) from the previous section's plan. The system would look up Location 35 and then discover that its closest neighboring locations were stops 34, 31 28, and 9, respectively by distance. After selecting 34 randomly from Truck 3 Stop 6 and moving it, the routes would now be as follows:

Stop	Truck 1	Truck 2	Truck 3	Truck 4	Truck 5
1	32	1	5	22	26
2	28	33	7	2	16
3	18	29	12	3	10
4	36	15	40	34	8
5	25	39	24	35	13
6	31	14	11	23	38
7	4	37	20		9
8	30	6	27		21
9		17	19		
10					

(remainder of page left blank for ease of reading)

Beginning Edge Move

Although there are random sequence moves that may operate on any chromosome/gene sequence, the beginning and ending of a sequence are of unique importance in this problem. This is because each truck must depart and return to the same distribution center. Therefore, an edge sequence at the beginning of one truck may be much more valuable and better fitted if at the beginning of another truck or reversed and placed at the end of another truck. It must be reversed because the contact point with the distribution sequence is the first element on the first truck, but then must become the last element on the second truck. Randomness determines whether this mutation takes on the beginning-to-beginning or the beginning-to-inverse-end mutation. Also, randomness determines whether this is a 2-gene or 3-gene mutation.

An example is shown as follows. Using the plan in the preceding section, assume that Truck 1 was randomly chosen as the starting location for a 2-sequence inverse mutation, with Truck 5 being the recipient. The outcome would look as follows:

Stop	Truck 1	Truck 2	Truck 3	Truck 4	Truck 5
1	18	1	5	22	26
2	36	33	7	2	16
3	25	29	12	3	10
4	31	15	40	34	8
5	4	39	24	35	13
6	30	14	11	23	38
7		37	20		9
8		6	27		21
9		17	19		28
10					32

(remainder of page left blank for ease of reading)

Ending Edge Move

This is the same idea as the preceding mutation, but it starts with the ending sequence, and moves it to another end or inverses it and moves it to the beginning of another truck. For example, using the routes in the preceding example, assume that a 3 stop sequence was randomly chosen from Truck 2 to be placed at the end of Truck 1; the following would be the result:

Stop	Truck 1	Truck 2	Truck 3	Truck 4	Truck 5
1	18	1	5	22	26
2	36	33	7	2	16
3	25	29	12	3	10
4	31	15	40	34	8
5	4	39	24	35	13
6	30	14	11	23	38
7	37		20		9
8	6		27		21
9	17		19		28
10					32

Beginning/End Edge Swap

The final edge-specific mutation is the beginning/end swap. In the preceding two mutations, the stop sequence was simply moved from one truck to a different truck. In this mutation, one sequence and another sequence are swapped with each other. For example, assume that a 2-sequence was randomly selected with Trucks 3 and 4 with Truck 3 being the starting truck. The route above would now be as follows:

Stop	Truck 1	Truck 2	Truck 3	Truck 4	Truck 5
1	18	1	23	22	26
2	36	33	35	2	16
3	25	29	12	3	10
4	31	15	40	34	8
5	4	39	24	7	13
6	30	14	11	5	38
7	6	37	20		9
8	17		27		21
9			19		28
10					32

(remainder of page left blank for ease of reading)

Appendix C

Sample Routing of Optimal Representative Week

SUMMARY

Day	Total	Mon	Tue	Wed	Thu	Fri
Trucks	31	6	5	7	7	6
Volume	65429	10223	11537	15192	15009	13468
Miles	6638	1388	1370	1196	1446	1238
Labor	254	52	55	35	58	53
ON Mins	538.5	37.5	87	124.5	121.5	168
Cost	\$10,995	\$2,272	\$2,340	\$1,719	\$2,472	\$2,192
\$/100ft3	\$16.80	\$22.22	\$20.28	\$11.31	\$16.47	\$16.28

Notes: Cost was calculated at a rate of 70 cents/mile and \$25/hour for labor.

ROUTING

Monday

Truck: 1

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		7:34:30				
1	28	Boston	MA	8:00:00	8:30:00	85	17		
2	70	Boston	MA	8:30:00	8:30:00	200	0		
3		DC/Wilmington	MA	8:55:30			17	0.06	
					Totals	285	34		

Monday

Truck: 2

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		7:48:00				
1	251	Billerica	MA	8:00:00	8:30:00	146	8		
2	188	Weston	MA	8:55:30	9:25:30	127	17		
3	32	Weston	MA	9:25:30	9:25:30	152	0		
4	201	Marlborough	MA	9:45:00	10:15:00	124	13		
5	20	Westborough	MA	10:30:00	11:00:00	111	10		
6	117	Auburndale	MA	11:33:00	12:03:00	161	22		
7	178	Brighton	MA	12:12:00	12:42:00	165	6		
8	90	West Roxbury	MA	12:51:00	13:21:00	187	6		
9	209	Boston	MA	13:34:30	14:04:30	246	9		
10	244	Boston	MA	14:07:30	14:37:30	163	2		

11	162	Boston	MA	14:40:30	15:10:30	106	2		
12		DC/Wilmington	MA	15:36:00			17	0.33	
					Totals	1688	112		

Monday

Overnight

Truck: 3

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		4:58:30				
1	237	Groton	CT	8:00:00	8:30:00	148	121		
2	207	Groton	CT	8:30:00	8:30:00	145	0		
3	76	New London	CT	8:37:30	9:07:30	631	5		
4	120	New London	CT	9:07:30	9:07:30	124	0		
5	40	Colchester	CT	9:43:30	10:13:30	213	24		
6	180	Bolton	CT	10:40:30	11:10:30	159	18		
7	45	East Hartford	CT	11:28:30	11:58:30	242	12		
8	80	Hartford	CT	12:04:30	12:34:30	110	4		
9	192	Hartford	CT	12:36:00	13:06:00	127	1		
10	168	Windsor Locks	CT	13:27:00	13:57:00	170	14		
11	15	Windsor	CT	14:04:30	14:34:30	157	5		
12	155	Manchester	CT	14:57:00	15:27:00	184	15		
13	217	Storrs Mansfield	CT	15:52:30	16:22:30	220	17		
14	67	Dayville	CT	17:01:30	17:31:30	102	26		
15		DC/Wilmington	MA	5:36:00			83	0.61	37.5
					Totals	2732	345		

Monday

Truck: 4

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		7:36:00				
1	255	Charlestown	MA	8:00:00	8:30:00	120	16		
2	23	Charlestown	MA	8:30:00	8:33:36	1676	0		
3		DC/Wilmington	MA	8:57:36			16	0.06	
					Totals	1796	32		

Monday

Truck: 5

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		6:13:30				
1	132	North Berwick	ME	8:00:00	8:30:00	244	71		
2	258	Burlington	VT	13:46:30	14:16:30	165	211		
3	175	Burlington	VT	14:16:30	14:16:30	160	0		
4		DC/Wilmington	MA	19:18:00			201	0.54	

					Totals	569	483
--	--	--	--	--	--------	-----	-----

Monday

Truck: 6

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		5:13:30				
1	223	Hartford	CT	8:00:00	8:30:00	144	111		
2	152	Hartford	CT	8:30:00	8:30:00	211	0		
3	58	Hartford	CT	8:30:00	8:30:00	183	0		
4	104	Hartford	CT	8:31:30	9:01:30	124	1		
5	14	Farmington	CT	9:16:30	10:00:52	1479	10		
6	241	Bethany	CT	10:45:52	11:15:52	153	30		
7	87	Wilton	CT	12:14:22	12:44:22	163	39		
8	85	Fairfield	CT	13:05:22	13:35:22	127	14		
9	250	Hamden	CT	14:15:52	14:45:52	167	27		
10	128	Middletown	CT	15:20:22	15:50:22	204	23		
11	109	Middletown	CT	15:50:22	15:50:22	198	0		
12		DC/Wilmington	MA	19:00:52			127	0.57	
					Totals	3153	382		

Tuesday

Truck: 1

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		5:07:30				
1	146	Hanover	NH	8:00:00	8:30:00	189	115		
2	182	Rumford	ME	12:06:00	12:36:00	140	144		
3	181	Parsonsfield	ME	14:37:30	15:07:30	243	81		
4	234	Dover	NH	16:22:30	16:52:30	237	50		
5		DC/Wilmington	MA	18:18:00			57	0.55	
					Totals	809	447		

Tuesday

Truck: 2

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		8:00:00				
1	7	DC/Wilmington	MA	8:00:00	8:30:00	491	0		
2	92	Boston	MA	8:55:30	9:25:30	226	17		
3	140	Boston	MA	9:27:00	9:57:00	136	1		
4	72	Boston	MA	9:57:00	9:57:00	199	0		
5	71	Cambridge	MA	10:00:00	10:30:00	164	2		
6	239	Cambridge	MA	10:30:00	10:30:00	122	0		
7	186	Cambridge	MA	10:31:30	11:01:30	131	1		

8	9	Cambridge	MA	11:01:30	11:01:30	182	0		
9	34	Boston	MA	11:06:00	11:36:00	301	3		
10	10	Boston	MA	11:36:00	11:36:00	203	0		
11	8	Lynn	MA	11:58:30	12:28:30	100	15		
12	66	Wenham	MA	12:55:30	13:25:30	197	18		
13	22	North Andover	MA	13:46:30	14:16:30	286	14		
14	4	North Andover	MA	14:16:30	14:16:30	111	0		
15		DC/Wilmington	MA	14:33:00			11	0.27	
					Totals	2849	82		

Tuesday

Truck: 3

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		7:48:00				
1	108	Billerica	MA	8:00:00	8:30:00	261	8		
2	78	Billerica	MA	8:30:00	8:30:00	146	0		
3	176	Bedford	MA	8:39:00	9:09:00	226	6		
4	212	Weston	MA	9:25:30	9:55:30	114	11		
5	26	Wellesley Hills	MA	10:03:00	10:33:00	409	5		
6	221	Chestnut Hill	MA	10:43:30	11:13:30	143	7		
7	42	Boston	MA	11:21:00	11:51:00	135	5		
8	122	Boston	MA	11:54:00	12:24:00	283	2		
9	33	Boston	MA	12:24:00	12:24:00	247	0		
10	51	Boston	MA	12:24:00	12:24:00	135	0		
11	163	Cambridge	MA	12:27:00	12:57:00	119	2		
12	211	Cambridge	MA	13:00:00	13:30:00	141	2		
13	187	Cambridge	MA	13:30:00	13:30:00	143	0		
14		DC/Wilmington	MA	13:55:30			17	0.26	
					Totals	2502	65		

Tuesday

Overnight

Truck: 4

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		4:21:00				
1	74	Bethany	CT	8:00:00	8:30:00	210	146		
2	105	Shelton	CT	8:55:30	9:25:30	572	17		
3	125	Stamford	CT	10:16:30	10:46:30	182	34		
4	107	Stamford	CT	10:46:30	10:46:30	106	0		
5	18	Greenwich	CT	11:01:30	11:31:30	394	10		
6	153	Stamford	CT	11:42:00	12:12:00	149	7		
7	224	Stamford	CT	12:12:00	12:12:00	276	0		

8	60	Norwalk	CT	12:27:00	12:57:00	110	10		
9	101	New London	CT	14:57:00	15:27:00	103	80		
10	49	New London	CT	15:27:00	15:27:00	256	0		
11	243	New London	CT	15:27:00	15:27:00	89	0		
12	29	Groton	CT	15:34:30	16:04:30	199	5		
13	68	Groton	CT	16:13:30	16:43:30	81	6		
14		DC/Wilmington	MA	5:48:00			123	0.64	87
					Totals	2727	438		

Tuesday

Truck: 5

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		5:15:00				
1	133	Enfield	CT	8:00:00	8:30:00	121	110		
2	179	Enfield	CT	8:30:00	8:30:00	135	0		
3	39	West Hartland	CT	9:22:30	9:52:30	257	35		
4	52	Farmington	CT	10:30:00	11:00:00	180	25		
5	177	Hartford	CT	11:13:30	11:43:30	164	9		
6	235	Hartford	CT	11:48:00	12:18:00	105	3		
7	216	Hartford	CT	12:18:00	12:48:00	263	0		
8	148	Hartford	CT	12:49:30	13:19:30	496	1		
9	254	Hartford	CT	13:22:30	13:52:30	111	2		
10	150	Middletown	CT	14:19:30	14:49:30	133	18		
11	57	Middletown	CT	14:49:30	14:49:30	180	0		
12	50	Glastonbury	CT	15:19:30	15:49:30	131	20		
13	121	Glastonbury	CT	15:49:30	15:49:30	243	0		
14	88	Manchester	CT	15:57:00	16:27:00	131	5		
15		DC/Wilmington	MA	19:12:00			110	0.58	
					Totals	2650	338		

Wednesday

Truck: 1

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		6:15:00				
1	65	Smithfield	RI	8:00:00	8:22:00	212	70		
2	62	Providence	RI	8:37:00	9:07:00	241	10		
3	27	East Providence	RI	9:11:30	9:58:30	1594	3		
4	12	East Providence	RI	9:58:30	10:45:30	608	0		
5	233	Exeter	RI	11:27:30	11:57:30	151	28		
6	205	Newport	RI	12:29:00	12:59:00	366	21		
7		DC/Wilmington	MA	15:11:00			88	0.37	
					Totals	3172	220		

Wednesday

Truck: 2

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		6:39:00				
1	200	Worcester	MA	8:00:00	8:22:00	159	54		
2	116	Marlborough	MA	8:50:30	9:20:30	164	19		
3	149	Framingham	MA	9:29:30	9:59:30	137	6		
4	260	Brighton	MA	10:26:30	10:56:30	132	18		
5	158	Boston	MA	11:05:30	11:35:30	170	6		
6	91	Boston	MA	11:35:30	11:35:30	250	0		
7	240	Boston	MA	11:40:00	12:10:00	174	3		
8	96	Boston	MA	12:10:00	12:10:00	106	0		
9	38	Medford	MA	12:19:00	12:10:00	143	6		
10	173	Medford	MA	12:19:00	12:10:00	96	0		
11		DC/Wilmington	MA	12:19:00			12	0.24	
					Totals	1531	124		

Wednesday

Truck: 3

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		7:51:00				
1	135	Tewksbury	MA	8:00:00	8:22:00	101	6		
2	138	Lowell	MA	8:35:30	9:05:30	159	9		
3	225	Keene	NH	10:41:30	11:11:30	129	64		
4	13	Keene	NH	11:11:30	11:11:30	117	0		
5	231	Northampton	MA	12:43:00	13:13:00	151	61		
6	81	Springfield	MA	13:41:30	14:11:30	140	19		
7	55	Windsor Locks	CT	14:34:00	15:04:00	112	15		
8	220	Windsor Locks	CT	15:04:00	15:04:00	98	0		
9	253	Windsor	CT	15:11:30	15:41:30	94	5		
10	167	Glastonbury	CT	16:07:00	16:37:00	140	17		
11	106	Storrs Mansfield	CT	17:10:00	17:40:00	186	22		
12	193	Storrs Mansfield	CT	17:40:00	17:40:00	118	0		
13		DC/Wilmington	MA	20:01:00			94	0.41	
					Totals	1545	312		

Wednesday

Truck: 4

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		7:46:30				
1	37	Woburn	MA	8:00:00	8:22:00	128	9		
2	1	Woburn	MA	8:22:00	8:22:00	333	0		

3		DC/Wilmington	MA	8:35:30		0	9	0.03	
					Totals	461	18		

Wednesday

Truck: 5

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
0		DC/Wilmington	MA		7:51:00	0	0		
1	6	Tewksbury	MA	8:00:00	8:22:00	2699	6		
2		DC/Wilmington	MA	8:31:00			6	0.03	
					Totals	2699	12		

Wednesday

Truck: 6

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		6:39:00				
1	43	Boston	MA	8:00:00	8:22:00	152	17		
2	206	Boston	MA	8:50:30	9:17:30	176	0		
3	77	Boston	MA	9:20:30	9:50:30	157	2		
4	139	Duxbury	MA	10:40:00	11:10:00	176	33		
5	24	Rockland	MA	11:34:00	12:36:00	2077	16		
6	73	Boston	MA	13:07:30	13:37:30	120	21		
7		DC/Wilmington	MA	14:04:30			18	0.31	
					Totals	2858	107		

Wednesday

Overnight

Truck: 7

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		4:24:00				
1	54	Hamden	CT	8:00:00	8:22:00	283	144		
2	261	West Haven	CT	8:35:30	9:05:30	130	9		
3	171	Norwalk	CT	9:52:00	10:22:00	376	31		
4	257	Stamford	CT	10:37:00	11:07:00	235	10		
5	53	Stamford	CT	11:17:30	11:47:30	112	7		
6	156	Stamford	CT	11:50:30	12:20:30	197	2		
7	48	Bethany	CT	13:31:00	14:01:00	396	47		
8	115	Plainville	CT	14:37:00	15:07:00	286	24		
9	246	Farmington	CT	15:14:30	15:44:30	129	5		
10	123	Farmington	CT	15:44:30	15:44:30	151	0		
11	248	Hartford	CT	15:59:30	16:29:30	149	10		
12	35	Hartford	CT	16:31:00	17:01:00	236	1		
13	238	East Hartford	CT	17:08:30	17:38:30	246	5		
14		DC/Wilmington	MA	6:20:30			108	0.08	124.5
					Totals	2926	403		

Thursday

Truck: 1

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		7:33:00				
1	202	Boston	MA	8:00:00	8:22:00	121	18		
2	103	Boston	MA	8:23:30	8:53:30	130	1		
3	245	Boston	MA	8:53:30	8:53:30	227	0		
4	189	Boston	MA	8:53:30	8:53:30	123	0		
5	79	Boston	MA	8:56:30	9:26:30	148	2		
6	69	Boston	MA	9:28:00	9:58:00	141	1		
7	118	Boston	MA	9:58:00	9:58:00	117	0		
8	183	Boston	MA	9:58:00	9:58:00	203	0		
9	97	Cambridge	MA	10:01:00	10:31:00	203	2		
10	131	Allston	MA	10:32:30	11:02:30	142	1		
11	190	Boston	MA	11:05:30	11:35:30	140	2		
12	124	Boston	MA	11:35:30	11:35:30	160	0		
13	215	Boston	MA	11:35:30	11:35:30	178	0		
14	95	Cambridge	MA	11:43:00	12:13:00	114	5		
15	46	Cambridge	MA	12:13:00	12:13:00	137	0		
16	31	Cambridge	MA	12:13:00	12:13:00	173	0		
17	159	Boston	MA	12:16:00	12:46:00	119	2		
18	184	Boston	MA	12:46:00	12:46:00	153	0		
19		DC/Wilmington	MA	13:11:30			17	0.24	
					Totals	2729	51		

Thursday

Truck: 2

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		5:31:30				
1	170	Portland	ME	8:00:00	8:22:00	188	99		
2	222	Portland	ME	8:22:00	8:22:00	105	0		
3	151	Portland	ME	8:22:00	8:22:00	105	0		
4	84	Westbrook	ME	8:32:30	9:02:30	175	7		
5	210	Cambridge	MA	11:46:00	12:16:00	401	109		
6	99	Cambridge	MA	12:19:00	12:49:00	172	2		
7	196	Winchester	MA	12:58:00	13:28:00	250	6		
8	21	Billerica	MA	13:41:30	14:22:30	1396	9		
9		DC/Wilmington	MA	14:34:30			8	0.38	
					Totals	2792	240		

Thursday

Truck: 3

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		5:13:30				
1	36	Hartford	CT	8:00:00	8:22:00	102	111		
2	102	Glastonbury	CT	8:34:00	9:04:00	168	8		
3	166	New London	CT	10:10:00	10:40:00	474	44		
4	93	Groton	CT	10:47:30	11:17:30	219	5		
5	160	Groton	CT	11:17:30	11:17:30	260	0		
6	259	North Dartmouth	MA	13:14:30	13:44:30	131	78		
7	154	Norton	MA	14:43:00	15:13:00	345	39		
8	232	Weymouth	MA	16:04:00	16:34:00	146	34		
9	185	Boston	MA	16:58:00	17:28:00	158	16		
10	141	Boston	MA	17:28:00	17:28:00	123	0		
11		DC/Wilmington	MA	17:55:00			18	0.53	
					Totals	2126	353		

Thursday

Truck: 4

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		7:31:30				
1	25	Waltham	MA	8:00:00	8:22:00	2614	19		
2	25	DC/Wilmington	MA	8:50:30			19	0.05	
					Totals	2614	38		

Thursday

Truck: 5

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		7:25:30				
1	113	Westford	MA	8:00:00	8:22:00	163	23		
2	195	Merrimack	NH	8:56:30	9:26:30	319	23		
3	199	Manchester	NH	9:44:30	10:14:30	197	12		
4	56	Manchester	NH	10:19:00	10:49:00	94	3		
5	218	Hanover	NH	12:50:30	13:20:30	193	81		
6	194	Hanover	NH	13:20:30	13:20:30	112	0		
7	86	Lebanon	NH	13:43:00	14:13:00	193	15		
8	136	Manchester	NH	16:04:00	16:34:00	257	74		
9		DC/Wilmington	MA	17:28:00			36	0.42	
					Totals	1528	267		

Thursday

Truck: 6

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		7:13:30				
1	219	Framingham	MA	8:00:00	8:22:00	137	31		

2	229	Marlborough	MA	8:31:00	9:01:00	180	6		
3	75	Weston	MA	9:20:30	9:50:30	122	13		
4	100	Weston	MA	9:50:30	9:50:30	265	0		
5	142	Weston	MA	9:50:30	9:50:30	195	0		
6	5	Reading	MA	10:20:30	10:50:30	206	20		
7		DC/Wilmington	MA	10:59:30			6	0.16	
					Totals	1105	76		

Thursday

Overnight

Truck: 7

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		4:43:30				
1	134	Meriden	CT	8:00:00	8:22:00	225	131		
2	61	Wallingford	CT	8:35:30	9:05:30	103	9		
3	63	New Haven	CT	9:28:00	9:58:00	112	15		
4	119	Bethany	CT	10:14:30	10:44:30	138	11		
5	256	Norwalk	CT	11:43:00	12:13:00	152	39		
6	197	Norwalk	CT	12:19:00	12:49:00	348	4		
7	137	New Canaan	CT	13:01:00	13:31:00	195	8		
8	147	Stamford	CT	13:44:30	14:14:30	173	9		
9	227	Ridgefield	CT	14:37:00	15:07:00	125	15		
10	247	Danbury	CT	15:17:30	15:47:30	166	7		
11	214	Danbury	CT	15:47:30	15:47:30	251	0		
12	127	Hartford	CT	17:19:00	17:49:00	127	61		
13		DC/Wilmington	MA	6:37:00			112	0.66	121.5
					Totals	2115	421		

Friday

Truck: 1

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		5:00:00				
1	204	New Britain	CT	8:00:00	8:30:00	423	120		
2	111	New Britain	CT	8:31:30	9:01:30	186	1		
3	213	Farmington	CT	9:15:00	9:45:00	154	9		
4	16	Hartford	CT	10:00:00	10:30:00	255	10		
5	145	Hartford	CT	10:31:30	11:01:30	122	1		
6	41	Granby	CT	11:31:30	12:01:30	163	20		
7	161	East Hartford	CT	12:33:00	13:03:00	237	21		
8	30	East Hartford	CT	13:03:00	13:03:00	205	0		
9	94	East Hartford	CT	13:03:00	13:03:00	150	0		
10	114	Tolland	CT	13:30:00	14:00:00	78	18		
11		DC/Wilmington	MA	16:22:30			95	0.47	

					Totals	1973	295		
--	--	--	--	--	--------	------	-----	--	--

Friday

Truck: 2

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		7:48:00				
1	2	Billerica	MA	8:00:00	8:30:00	556	8		
2	203	Fitchburg	MA	9:28:30	9:58:30	444	39		
3	89	Clinton	MA	10:24:00	10:54:00	390	17		
4	130	Webster	MA	11:42:00	12:12:00	336	32		
5	198	Westborough	MA	12:51:00	13:21:00	102	26		
6	228	Southborough	MA	13:31:30	14:01:30	119	7		
7	249	Framingham	MA	14:13:30	14:43:30	106	8		
8	174	Waltham	MA	15:06:00	15:36:00	182	15		
9	165	Weston	MA	15:42:00	16:12:00	128	4		
10	191	Waltham	MA	16:18:00	16:48:00	131	4		
11		DC/Wilmington	MA	17:16:30			19	0.39	
					Totals	2494	179		

Friday

Truck: 3

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		7:51:00				
1	230	Reading	MA	8:00:00	8:30:00	132	6		
2	126	Woburn	MA	8:37:30	9:07:30	195	5		
3	59	Charlestown	MA	9:22:30	9:52:30	187	10		
4	208	Boston	MA	9:55:30	10:25:30	140	2		
5	47	Boston	MA	10:28:30	10:58:30	121	2		
6	82	Boston	MA	11:01:30	11:31:30	161	2		
7	236	Boston	MA	11:33:00	12:03:00	221	1		
8	169	Boston	MA	12:04:30	12:34:30	202	1		
9	242	Cambridge	MA	12:43:30	13:13:30	109	6		
10		DC/Wilmington	MA	13:39:00			17	0.24	
					Totals	1468	52		

Friday

Truck: 4

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		7:30:00				
1	252	Salem	MA	8:00:00	8:30:00	181	20		
2	110	Portland	ME	10:52:30	11:22:30	177	95		
3	3	Lawrence	MA	13:37:30	14:07:30	903	90		

4		DC/Wilmington	MA	14:24:00			11	0.29	
					Totals	1261	216		

Friday

Overnight

Truck: 5

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		3:43:30				
1	144	Danbury	CT	8:00:00	8:30:00	147	171		
2	172	Danbury	CT	8:34:30	9:04:30	261	3		
3	83	Ridgefield	CT	9:15:00	9:45:00	172	7		
4	19	Stamford	CT	10:21:00	10:51:00	896	24		
5	157	Old Greenwich	CT	10:55:30	11:25:30	126	3		
6	129	Stamford	CT	11:30:00	12:00:00	88	3		
7	64	Fairfield	CT	12:27:00	12:57:00	143	18		
8	112	Stratford	CT	13:19:30	13:49:30	181	15		
9	226	Stratford	CT	13:49:30	13:49:30	187	0		
10	164	Bethany	CT	14:25:30	14:55:30	149	24		
11	98	Bethany	CT	14:55:30	14:55:30	98	0		
12	17	New London	CT	16:22:30	16:52:30	343	58		
13	143	New London	CT	16:52:30	16:52:30	164	0		
14	44	Groton	CT	17:00:00	17:30:00	164	5		
15		DC/Wilmington	MA	20:31:30			121	0.70	168
					Totals	3119	452		

Friday

Truck: 6

Stop	OrderID	Location		Arrive	Depart	Vol. ft3	Miles	Labor	ONMins
		DC/Wilmington	MA		7:27:00				
1	11	Weston	MA	8:00:00	9:34:35	3153	22		
2		DC/Wilmington	MA	10:07:35			22	0.11	
					Totals	3153	44		

Appendix D
Example of CSV Data Output by GS490
(with added time column calculations)

Monday, 6 trucks, Score: 1355

Truck	stop	orderid	from	to	travelmin	cumtravel	volume	cumvolume	unloadmin	cumunload	miles	cummiles	overntmin	Depart	Arrive	Complete	DOTComplete	
0	0	251	-1	13	12	12	146	146	30	30	8	8		8:00	7:48:00	8:00:00	8:22:00	8:22:00
0	1	251	13	-1	12	24	0	146	0	30	8	16		8:22:00	8:34:00	8:34:00	8:34:00	8:34:00
1	0	70	-1	26	25.5	25.5	200	200	30	30	17	17		8:00	7:34:30	8:00:00	8:22:00	8:22:00
1	1	28	26	26	0	25.5	85	285	0	30	0	17		8:22:00	8:22:00	8:22:00	8:22:00	8:22:00
1	2	209	26	27	1.5	27	246	531	30	60	1	18		8:22:00	8:23:30	8:53:30	8:53:30	8:53:30
1	3	244	27	42	3	30	163	694	30	90	2	20		8:53:30	8:56:30	9:26:30	9:26:30	9:26:30
1	4	162	42	28	3	33	106	800	30	120	2	22		9:26:30	9:29:30	9:59:30	9:59:30	9:59:30
1	5	178	28	35	9	42	165	965	30	150	6	28		9:59:30	10:08:30	10:38:30	10:38:30	10:38:30
1	6	90	35	33	9	51	187	1152	30	180	6	34		10:38:30	10:47:30	11:17:30	11:17:30	11:17:30
1	7	117	33	48	13.5	64.5	161	1313	30	210	9	43		11:17:30	11:31:00	12:01:00	12:01:00	12:01:00
1	8	20	48	6	33	97.5	111	1424	30	240	22	65		12:01:00	12:34:00	13:04:00	13:04:00	13:04:00
1	9	201	6	10	15	112.5	124	1548	30	270	10	75		13:04:00	13:19:00	13:49:00	13:49:00	13:49:00
1	10	188	10	51	19.5	132	127	1675	30	300	13	88		13:49:00	14:08:30	14:38:30	14:38:30	14:38:30
1	11	32	51	51	0	132	152	1827	0	300	0	88		14:38:30	14:38:30	14:38:30	14:38:30	14:38:30
1	12	32	51	-1	33	165	0	1827	0	300	22	110		14:38:30	15:11:30	15:11:30	15:11:30	15:11:30
2	0	175	-1	72	301.5	301.5	160	160	30	30	201	201		8:00	2:58:30	8:00:00	8:22:00	8:22:00
2	1	258	72	72	0	301.5	165	325	0	30	0	201		8:22:00	8:22:00	8:22:00	8:22:00	8:22:00
2	2	132	72	67	316.5	618	244	569	30	60	211	412		8:22:00	13:38:30	14:08:30	14:08:30	14:08:30
2	3	132	67	-1	106.5	724.5	0	569	0	60	71	483	64.5	14:08:30	15:55:00	15:55:00	15:55:00	1:55:00
3	0	67	-1	94	124.5	124.5	102	102	30	30	83	83		8:00	5:55:30	8:00:00	8:22:00	8:22:00
3	1	237	94	97	75	199.5	148	250	30	60	50	133		8:22:00	9:37:00	10:07:00	10:07:00	10:07:00
3	2	207	97	97	0	199.5	145	395	0	60	0	133		10:07:00	10:07:00	10:07:00	10:07:00	10:07:00
3	3	76	97	96	7.5	207	631	1026	30	90	5	138		10:07:00	10:14:30	10:44:30	10:44:30	10:44:30
3	4	120	96	96	0	207	124	1150	0	90	0	138		10:44:30	10:44:30	10:44:30	10:44:30	10:44:30
3	5	40	96	99	36	243	213	1363	30	120	24	162		10:44:30	11:20:30	11:50:30	11:50:30	11:50:30

3	6	223	99	93	40.5	283.5	144	1507	30	150	27	189	11:50:30	12:31:00	13:01:00	13:01:00
3	7	152	93	93	0	283.5	211	1718	0	150	0	189	13:01:00	13:01:00	13:01:00	13:01:00
3	8	58	93	93	0	283.5	183	1901	0	150	0	189	13:01:00	13:01:00	13:01:00	13:01:00
3	9	80	93	86	1.5	285	110	2011	30	180	1	190	13:01:00	13:02:30	13:32:30	13:32:30
3	10	192	86	87	1.5	286.5	127	2138	30	210	1	191	13:32:30	13:34:00	14:04:00	14:04:00
3	11	45	87	90	4.5	291	242	2380	30	240	3	194	14:04:00	14:08:30	14:38:30	14:38:30
3	12	155	90	76	9	300	184	2564	30	270	6	200	14:38:30	14:47:30	15:17:30	15:17:30
3	13	180	76	77	9	309	159	2723	30	300	6	206	15:17:30	15:26:30	15:56:30	15:56:30
3	14	217	77	95	21	330	220	2943	30	330	14	220	15:56:30	16:17:30	16:47:30	16:47:30
3	15	217	95	-1	141	471	0	2943	0	330	94	314	16:47:30	19:08:30	19:08:30	19:08:30
4	0	23	-1	32	24	24	1676	1676	50.28	50.28	16	16	8:00	7:36:00	8:00:00	8:22:00
4	1	255	32	32	0	24	120	1796	50.28	100.56	0	16	8:22:00	8:22:00	9:12:00	9:12:00
4	2	255	32	-1	24	48	0	1796	0	100.56	16	32	9:12:00	9:36:00	9:36:00	9:36:00
5	0	168	-1	85	172.5	172.5	170	170	30	30	115	115	8:00	5:07:30	8:00:00	8:22:00
5	1	15	85	84	7.5	180	157	327	30	60	5	120	8:22:00	8:29:30	8:59:30	8:59:30
5	2	104	84	92	15	195	124	451	30	90	10	130	8:59:30	9:14:30	9:44:30	9:44:30
5	3	14	92	73	15	210	1479	1930	44.37	134.37	10	140	9:44:30	9:59:30	10:43:30	10:43:30
5	4	241	73	107	45	255	153	2083	30	164.37	30	170	10:43:30	11:28:30	11:58:30	11:58:30
5	5	85	107	112	42	297	127	2210	30	194.37	28	198	11:58:30	12:40:30	13:10:30	13:10:30
5	6	87	112	119	21	318	163	2373	30	224.37	14	212	13:10:30	13:31:30	14:01:30	14:01:30
5	7	250	119	106	57	375	167	2540	30	254.37	38	250	14:01:30	14:58:30	15:28:30	15:28:30
5	8	109	106	101	34.5	409.5	198	2738	30	284.37	23	273	15:28:30	16:03:00	16:33:00	16:33:00
5	9	128	101	101	0	409.5	204	2942	0	284.37	0	273	16:33:00	16:33:00	16:33:00	16:33:00
5	10	128	101	-1	190.5	600	0	2942	0	284.37	127	400	44.37	16:33:00	19:43:30	19:43:30

Fit: 1355

Dist: 1355

Wednesday, 7 trucks, Score: 1196

truck	stop	orderid	from	to	travelmin	cumtravel	volume	cumvolume	unloadmin	cumunload	miles	cummiles	overntmin	Depart	Arrive	Complete	DOTComplete
0	0	65	-1	58	105	105	212	212	30	30	70	70	8:00	6:15:00	8:00:00	8:22:00	8:22:00

0	1	62	58	56	15	120	241	453	30	60	10	80		8:22:00	8:37:00	9:07:00	9:07:00
0	2	27	56	57	4.5	125	1594	2047	47.8	108	3	83		9:07:00	9:11:30	9:58:30	9:58:30
0	3	12	57	57	0	125	608	2655	47.8	156	0	83		9:58:30	9:58:30	10:45:30	10:45:30
0	4	233	57	54	42	167	151	2806	30	186	28	111		10:45:30	11:27:30	11:57:30	11:57:30
0	5	205	54	55	31.5	198	366	3172	30	216	21	132		11:57:30	12:29:00	12:59:00	12:59:00
0	6	205	55	-1	132	330	0	3172	0	216	88	220		12:59:00	15:11:00	15:11:00	15:11:00
1	0	200	-1	7	81	81	159	159	30	30	54	54	8:00	6:39:00	8:00:00	8:22:00	8:22:00
1	1	116	7	10	28.5	110	164	323	30	60	19	73		8:22:00	8:50:30	9:20:30	9:20:30
1	2	149	10	8	9	119	137	460	30	90	6	79		9:20:30	9:29:30	9:59:30	9:59:30
1	3	260	8	35	27	146	132	592	30	120	18	97		9:59:30	10:26:30	10:56:30	10:56:30
1	4	158	35	29	9	155	170	762	30	150	6	103		10:56:30	11:05:30	11:35:30	11:35:30
1	5	91	29	29	0	155	250	1012	0	150	0	103		11:35:30	11:35:30	11:35:30	11:35:30
1	6	240	29	28	4.5	159	174	1186	30	180	3	106		11:35:30	11:40:00	12:10:00	12:10:00
1	7	96	28	28	0	159	106	1292	0	180	0	106		12:10:00	12:10:00	12:10:00	12:10:00
1	8	38	28	39	9	168	143	1435	30	210	6	112		12:10:00	12:19:00	12:49:00	12:49:00
1	9	173	39	39	0	168	96	1531	0	210	0	112		12:10:00	12:19:00	12:49:00	12:49:00
1	10	173	39	-1	18	186	0	1531	0	210	12	124		12:10:00	12:19:00	12:49:00	12:49:00
2	0	43	-1	26	25.5	25.5	152	152	30	30	17	17	8:00	6:39:00	8:00:00	8:22:00	8:22:00
2	1	206	26	26	0	25.5	176	328	0	30	0	17		8:22:00	8:50:30	9:20:30	9:20:30
2	2	77	26	42	3	28.5	157	485	30	60	2	19		9:17:30	9:20:30	9:50:30	9:50:30
2	3	139	42	44	49.5	78	176	661	30	90	33	52		9:50:30	10:40:00	11:10:00	11:10:00
2	4	24	44	45	24	102	2077	2738	62.3	152	16	68		11:10:00	11:34:00	12:36:00	12:36:00
2	5	73	45	27	31.5	134	120	2858	30	182	21	89		12:36:00	13:07:30	13:37:30	13:37:30
2	6	73	27	-1	27	161	0	2858	0	182	18	107		13:37:30	14:04:30	14:04:30	14:04:30
3	0	135	-1	18	9	9	101	101	30	30	6	6	8:00	7:51:00	8:00:00	8:22:00	8:22:00
3	1	138	18	16	13.5	22.5	159	260	30	60	9	15		8:22:00	8:35:30	9:05:30	9:05:30
3	2	225	16	63	96	119	129	389	30	90	64	79		9:05:30	10:41:30	11:11:30	11:11:30
3	3	13	63	63	0	119	117	506	0	90	0	79		11:11:30	11:11:30	11:11:30	11:11:30
3	4	231	63	1	91.5	210	151	657	30	120	61	140		11:11:30	12:43:00	13:13:00	13:13:00
3	5	81	1	2	28.5	239	140	797	30	150	19	159		13:13:00	13:41:30	14:11:30	14:11:30
3	6	55	2	85	22.5	261	112	909	30	180	15	174		14:11:30	14:34:00	15:04:00	15:04:00
3	7	220	85	85	0	261	98	1007	0	180	0	174		15:04:00	15:04:00	15:04:00	15:04:00

3	8	253	85	84	7.5	269	94	1101	30	210	5	179		15:04:00	15:11:30	15:41:30	15:41:30
3	9	167	84	74	25.5	294	140	1241	30	240	17	196		15:41:30	16:07:00	16:37:00	16:37:00
3	10	106	74	95	33	327	186	1427	30	270	22	218		16:37:00	17:10:00	17:40:00	17:40:00
3	11	193	95	95	0	327	118	1545	0	270	0	218		17:40:00	17:40:00	17:40:00	17:40:00
3	12	193	95	-1	141	468	0	1545	0	270	94	312					
4	0	54	-1	106	216	216	283	283	30	30	144	144	8:00	4:24:00	8:00:00	8:22:00	8:22:00
4	1	261	106	105	13.5	230	130	413	30	60	9	153		8:22:00	8:35:30	9:05:30	9:05:30
4	2	171	105	116	46.5	276	376	789	30	90	31	184		9:05:30	9:52:00	10:22:00	10:22:00
4	3	257	116	120	15	291	235	1024	30	120	10	194		10:22:00	10:37:00	11:07:00	11:07:00
4	4	53	120	123	10.5	302	112	1136	30	150	7	201		11:07:00	11:17:30	11:47:30	11:47:30
4	5	156	123	122	3	305	197	1333	30	180	2	203		11:47:30	11:50:30	12:20:30	12:20:30
4	6	48	122	107	70.5	375	396	1729	30	210	47	250		12:20:30	13:31:00	14:01:00	14:01:00
4	7	115	107	80	36	411	286	2015	30	240	24	274		14:01:00	14:37:00	15:07:00	15:07:00
4	8	246	80	73	7.5	419	129	2144	30	270	5	279		15:07:00	15:14:30	15:44:30	15:44:30
4	9	123	73	73	0	419	151	2295	0	270	0	279		15:44:30	15:44:30	15:44:30	15:44:30
4	10	248	73	88	15	434	149	2444	30	300	10	289		15:44:30	15:59:30	16:29:30	16:29:30
4	11	35	88	92	1.5	435	236	2680	30	330	1	290		16:29:30	16:31:00	17:01:00	17:01:00
4	12	238	92	90	7.5	443	246	2926	30	360	5	295		17:01:00	17:08:30	17:38:30	17:38:30
4	13	238	90	-1	162	605	0	2926	0	360	108	403	125	17:38:30	20:20:30	20:20:30	6:20:30
5	0	37	-1	12	13.5	13.5	128	128	30	30	9	9	8:00	7:46:30	8:00:00	8:22:00	8:22:00
5	1	1	12	12	0	13.5	333	461	0	30	0	9		8:22:00	8:22:00	8:22:00	8:22:00
5	2	1	12	-1	13.5	27	0	461	0	30	9	18		8:22:00	8:35:30	8:35:30	8:35:30
6	0	6	-1	18	9	9	2699	2699	81	81	6	6	8:00	7:51:00	8:00:00	8:22:00	8:22:00
6	1	6	18	-1	9	18	0	2699	0	81	6	12		8:22:00	8:31:00	8:31:00	8:31:00

Fit: 1196

Dist: 1196