Strategic Load-Planning for Less-Than-Truckload Trucking

Bruce Hoppe, Erica Zimmer Klampfl,
Cassandra McZeal, and Jennifer Rich

CRPC-TR99812-S
November 1999

Center for Research on Parallel Computation
Rice University
6100 South Main Street
CRPC - MS 41
Houston, TX 77005

Submitted November 1999

Strategic Load Planning for Less-Than-Truckload Trucking

Bruce Hoppe
Erica Zimmer Klampfl
Cassandra McZeal
Jennifer Rich

November 10, 1998

1 Introduction

Less-than-truckload trucking represents a portion of the motor carrier industry in which the shipments to be sent on trucks do not completely fill a 45,000 lb. volume tractor-trailer [4]. Typically, the freight in each shipment weighs under 10,000 lbs., with a large majority falling under 1000 lbs. Since each shipment does not fill a truck, significant savings can be achieved by consolidating shipments into loads at regional terminals and transporting these loads from terminal to terminal. The goal of the strategic load planning problem is to determine how to route the flow of consolidated loads from origin terminals to destination terminals cost effectively and allowing for certain service standards. The actual gathering of these shipments at the origin terminal and the distribution of them from the destination terminal is handled in a separate problem commonly referred to as the pick-up and delivery problem. This overall method of distribution requires a network of terminals, the design of which is closely related to many classic network design problems. We have built a software system which, when given the appropriate data concerning a company's needs and past routing decisions, will build a network to solve the strategic load planning problem. The empirical results, based on real data from trucking companies, indicates that our system does a very credible job of building an efficient network.

In section two of this paper, we define our formulation of the strategic load planning problem. A mathematical model is provided for clarity, but is not essential to the methods discussed in the rest of the paper. In section three, we discuss the implemented solution strategy, including both a network–building procedure and a final–stage improvement heuristic. Section four provides results obtained from data provided by real trucking companies. Finally, in section five we draw some conclusions and briefly examine some possibilities for future work.

2 Problem Formulation

2.1 Simplifying Assumptions

In order to model the strategic load planning problem, certain simplifying assumptions are made. First, we assume that all freight moves via trucks. No airplanes, boats or trains are available to ship the loads. We also assume that the trailers are homogeneous. That is, we have one type of trailer with one fixed capacity. We assume an unlimited supply of trailers are available for transporting the loads, and they are all owned by the company. In order to facilitate higher–level decisions beyond the scope of this particular problem, we allow the number of trailers used to ship loads to be fractional. We do not consider driver feasibility when creating the load plan. For example, direct service may be provided between two terminals whose distance far exceeds the number of miles one driver could traverse nonstop. Additionally, we assume that a driver will be available at any terminal to take a load wherever the load plan dictates.

2.2 Problem Description

The strategic load planning problem, or SLP, is based on data obtained from an analysis of a company's past and projected freight distribution needs. This analysis yields a set of loads, referred to as commodities, each with an origin and destination terminal and a quantity. These commodities are aggregated, meaning that a given origin/destination pair uniquely identifies a commodity. A commodity with zero freight simply indicates that no freight is shipped from that commodity's origin to its destination. Thus, commodities represent the expected shipment needs of the company. If the origin to destination path for a given commodity contains intermediate terminals, then the commmodity's freight is said to be "handled" at these terminals. Each terminal has a capacity in terms of the quantity of freight handled per day, as well as a handling cost per unit of freight and a handling time.

The arcs of the overall network are defined by the set of potential direct services between terminals. A potential direct service is an ordered pair of terminals and refers to the possibility of routing freight from one terminal to another with no intermediate stops at other terminals. Each direct service has a unit cost per amount of freight routed along the arc and a minimum frequency requirement. If freight is routed along a direct service, then the minimum frequency refers to the minimum number of trailers that the company must send along that arc throughout a day. If no freight is routed along the direct service, then the minimum frequency restriction does not apply. Finally, the company may also set certain service standards for each origin/destination pair. These standards refer to the maximum time permitted to transport a commodity from its origin to its destination. These restrictions limit the lengths of the routes.

The solution to the strategic load planning problem must provide two key pieces of information. First, it must reveal the status of each potential direct service. This indicates that either freight is routed along a given direct service (status in) or it is not (status out).

Second, a load plan providing the path for each commodity from its origin to its destination is required. This load plan must be tree-based so that given any commodity at a current terminal location and its known destination terminal, the next terminal on the commodity's path is uniquely defined. The set of terminals, along with the status in directs, define the solution network. Other restrictions on the solution include that there be enough trailers traveling along each status in direct service to carry the amount of freight being sent along that arc, as well as to meet the minimum frequency requirement. A terminal may not exceed its handling capacity for freight and must have zero net change in trailers, which may require the movement of empty trailers. The path for each commodity may not exceed the service standard. Finally, the goal is to minimize the sum of the cost of sending trailers along direct services and the cost of handling freight at the terminals.

2.3 Mathematical Model

The SLP can be formulated as a mixed integer programming problem in the following manner. The set of terminals for the strategic load planning problem is denoted by T. For each t in T, we have a handling cost, HC_t, and a handling time, HT_t. Also for each terminal, there is a capacity on the number of trailers of freight that it can handle each day, CAP_t. The set of commodities to be shipped is denoted by C and has elements kd where k is the origination terminal and d is its destination terminal (recall, commodities are unique origin/destination pairs). Within T, we also define the set $DEST$ to be the set of destination terminals for the set of commodities C. The parameter Q_{kd} is the number of trailers, possibly fractional, needed to ship commodity kd. The amount of time advertised to ship commodity kd from its origin terminal to its destination terminal is the service standard, ST_{kd}. The set of potential direct services is denoted by $DIRS$. For each ij in $DIRS$, TC_{ij} is the cost of sending one trailer from terminal i to terminal j with transit time TT_{ij}. The final parameter, MF_{ij}, refers to the minimum number of trailers that must be sent along arc ij throughout a day, if freight is allowed to move along ij.

The nodes in the network are enumerated by ikd, with $i \in T$ and $kd \in C$. The nodes created at each of the terminals for each commodity are connected by the decision variable f_{ijkd} representing the assignment of $ij \in DIRS$ to the freight path for $kd \in C$. The variable f_{ijkd} is 1 if direct ij is used to ship commodity k to its destination terminal d. These variables define the freight path for each commodity. The decision variable n_{ijd} is 1 if the next transfer at i is j for a commodity whose final destination is d. These variables determine a load plan that depends only on the destination terminal for a commodity. The variable v_{ij} is the number of trailers sent on direct ij. The strategic load planning problem is stated as:

$$
\begin{aligned}
&\text{Minimize} \quad \sum_{ij\in DIRS} TC_{ij}v_{ij} + \sum_{kd\in C,k,d\neq j}\ \sum_{ij\in DIRS} HC_j Q_{kd} f_{ijkd} \\
&\text{subject to}
\end{aligned}
$$

$$\sum_{ij\in DIRS} f_{ijkd} - \sum_{ji\in DIRS} f_{jikd} = \begin{cases} -1 & \text{if } i = d \\ 0 & \text{if } i \neq k, d \end{cases} \qquad i \in T, \quad kd \in C \tag{1}$$

$$\sum_{kd\in C,k,d\neq t} (Q_{kd} \sum_{it\in DIRS} f_{itkd}) \leq CAP_t \qquad t \in T \tag{2}$$

$$\sum_{ij\in DIRS} (TT_{ij} + HT_j) f_{ijkd} \leq ST_{kd} + HT_d \qquad kd \in C \tag{3}$$

$$f_{ijkd} \leq n_{ijd} \qquad ij \in DIRS, \quad kd \in C \tag{4}$$

$$\sum_{tj\in DIRS} n_{tjd} \leq 1 \qquad d \in DEST, \quad t \in T, t \neq d \tag{5}$$

$$n_{djd} = 0 \qquad d \in DEST, \quad dj \in DIRS \tag{6}$$

$$\sum_{kd\in C} Q_{kd} f_{ijkd} \leq v_{ij} \qquad ij \in DIRS \tag{7}$$

$$\sum_{it\in DIRS} v_{it} - \sum_{ti\in DIRS} v_{ti} = 0 \qquad t \in T \tag{8}$$

$$v_{ij} \geq MF_{ij} n_{ijd} \qquad ij \in DIRS, \quad d \in DEST \tag{9}$$

$$n_{ijd}, f_{ijkd} \in \{0, 1\} \qquad ij \in DIRS, kd \in C \tag{10}$$

The objective is to minimize total cost. The costs incurred are the transit costs per trailer on each direct service provided, in addition to the cost of handling freight at each of the intermediate handling terminals on a commodity's freight path. The first set of constraints are the flow conservation constraints for each ikd in the network with the exception that a demand of 1 be met at node dkd for all $kd \in C$. The second set of constraints are the terminal capacity constraints. The constraints specify that each terminal handle no more than its capacity. The third set is the service standard constraints which insure that the handling time and transit time for a commodity's freight path to its destination terminal is no more than the advertised time for arrival at the destination. The fourth set of constraints link the freight path variables to the load path variables so that if the next destination from i is not j for a commodity destined for d, then f_{ijkd} is forced to be zero.

Constraints (5) and (6) insure that our load plan is tree-based and rooted in the destination terminals. Constraint set (5) allows no more than one next transfer from any terminal t for a specific destination d. The next set eliminates any next transfers from a commodity's destination terminal. Constraints (7)-(9) are related to the trailers. Constraint set (7) is the set of trailer capacity constraints which ensure that commodities shipped on a trailer do not exceed trailer capacity. The next set ensures that trailers balance at each terminal, and (9)

specifies that the number of trailers shipping commodities on a direct service is at least the minimum frequency. Finally, the number of trailers is at least zero, and the freight path and load plan decision variables are binary.

Clearly, the mathematical formulation of the problem can become very large. The number of integer variables alone is on the order of $|DIRS| * |C|$. Since both the number of directs and the number of commodities are bounded by the square of the number of terminals, the number of integer variables is on the order of $|T|^4$. Given the size and difficulty of the integer programming formulation for any realistic data set, it is reasonable to use an approximate solution method, as is done in this paper.

3 Solution Strategy

Our solution strategy can be divided into three major phases. The first phase involves pruning the overall network of possible direct services. Our goal in this phase is to eliminate the more unlikely direct services while also maintaining feasibility. The second stage is an iterative network building phase. We use a modified version of Balakrishnan, Magnanti, and Wong's [2] dual–ascent procedure for uncapacitated network design to build the network at each iteration. Also, at each iteration we modify the network cost vectors in order to heuristically encourage solutions that adhere more closely to the requirements of the strategic load planning problem. In the last stage of the algorithm, we apply an add/drop procedure to the final solution from the iterative phase. The add/drop procedure systematically adds and drops direct services in order to find local improvements in the solution.

3.1 Pruning Techniques

To understand the necessity of pruning the overall network, consider an instance with 50 terminals, a complete, directed graph such that each arc is a potential direct service, and the requirement that some quantity of freight travel between every pair of terminals. A complete graph, in this case, implies a network with 2,450 arcs. The number of commodities is the same. Thus, the number of network flow variables is well over 6 million. Other than eliminating the commodities with zero freight, we can not reduce the number of commodities. We can, however, eliminate arcs as a means of reducing the problem size. To ensure the feasibility of the problem, we require the existence of a known feasible solution which we call the historical load plan, or HLP. The HLP is a complete load plan, supplied as part of the original data, and represents a solution previously used by the company. By not eliminating any arcs belonging to the HLP, the known solution remains feasible, and thus, a valid upper bound on the optimal solution.

By pruning arcs which we determine are unlikely to be in the optimal load plan, we can reduce the problem size significantly. Two factors which we consider in pruning the arcs are the type of terminals at an arc's endpoints and the amount of freight that would be sent along that arc if all commodities were sent along direct services from their origins to their

destinations. A terminal's type is either an *end–of–line (EOL)* or a *breakbulk*. An end–of–line terminal is one which is not permitted to handle any freight. That is, freight may either originate or terminate at the terminal, but it may not be an intermediate terminal on a commodity's path. A breakbulk terminal is permitted to handle intermediate transfers of freight. Freight originating and terminating at a terminal is not considered to be handled. Given this knowledge about the terminals, we know exactly how much freight would ever travel on an arc from one EOL to another – only the commodity corresponding to that direct service. If this commodity's quantity is less than the low freight parameter and the arc is not in the HLP, the arc is eliminated from the network. In general, as the low freight parameter increases, so does the number of arcs that are pruned.

Given additional data, further reductions are possible. If a company can provide an historical load summary relating how many loaded trailers are used over a certain time frame, this information can be used to prune less frequently used direct services. The real world data is based on the use of pups, which translate into half of a trailer, and vans, which translate into one trailer. If the number of trailers previously sent along an arc is less than the predefined parameter and is not in the HLP, then the arc is pruned. Also in this pruning technique, as this parameter indicating the minimum tractors permitted increases, in general the number of pruned arcs increases. While there does exist the risk that optimal solutions are eliminated by removing arcs using these two pruning techniques, we rely on the small likelihood of such a case so long as the values of these parameters are relatively small.

3.2 The Uncapacitated Network Design Problem

The UND and related UND problems

An uncapacitated network design (UND) problem consists of a set of nodes, a set of uncapacitated arcs, and a required flow of one unit that must be routed between specified pairs of nodes. Each arc has a fixed cost for using the arc and a per unit cost for the amount of flow along the arc. The optimization problem is to select the subset of arcs along which to route the required flow that minimizes the total cost. While this is an $\mathcal{NP}$–hard problem, there exists several approximate solution procedures which have succeeded in finding good solutions for large scale problems of this type. A survey of network design problems and solution methods can be found in a paper written by Balakrishnan, Magnanti, and Mirchandani[1].

The general case of the UND problem which we will consider has directed arcs ij in the set of arcs A and is defined over the set of nodes V. There is a set of commodities K which define the sets of pairs which must have a flow from the origin to the destination of any given commodity k (denoted $O(k)$ and $D(k)$, respectively). Unit costs are commodity dependent; thus, c_{ij}^k is the per unit cost of routing commodity k along arc ij. Fixed costs, however, are incurred if any commodity is routed along an arc. Thus, F_{ij} denotes the fixed cost of using the arc ij for any commodity. Given these cost parameters, only two types of variables are needed. The first variable is the flow variable x_{ij}^k, which is a variable indicating the fraction of commodity $k \in K$ which is routed along arc $ij \in A$. The second variable is the binary

variable y_{ij} indicating if the arc $ij \in A$ is used ($y_{ij} = 1$), or not ($y_{ij} = 0$). Thus, the primal UND problem (P) can be written as follows:

$$
\begin{array}{lll}
\text{Minimize} & \displaystyle\sum_{ij\in A}\sum_{k\in K} c_{ij}^k x_{ij}^k + \sum_{ij\in A} F_{ij} y_{ij} & \\
\text{subject to} & & \\
& \displaystyle\sum_{j\in V} x_{ji}^k - \sum_{j\in V} x_{ij}^k = \begin{cases} 1 & \text{if } i = D(k) \\ 0 & \text{if } i \neq O(k), D(k) \end{cases} & k \in K, i \in V \\
& x_{ij}^k \leq y_{ij} & ij \in A, k \in K \\
& x_{ij}^k \geq 0 & ij \in A, k \in K \\
& y_{ij} \in \{0,1\} & ij \in A.
\end{array}
$$

Note that in this formulation the flow variables x_{ij}^k are permitted to be free variables even though they will always take on binary values (i. e. either none of a commodity or all of a commodity is routed along an arc). This is due to the unit demand assumed in the network flow constraints.

Given a strategic load planning problem, we can define a *related UND problem* by ignoring the capacity restrictions, service requirements, and empty trailer balancing and by approximating the minimum frequency restrictions with the fixed cost parameter of the UND model. More specifically, the related UND model is constructed from an SLP problem containing the following data. First, the amount of freight associated with a commodity $k \in K$ is represented by $frt(k)$. Similarly, for each arc ij in the SLP network, transit cost on the arc is defined to be $transcost(ij)$, and the minimum frequency on the arc is defined to be $minfreq(ij)$. For each terminal i in the set of terminals, $handcost(i)$ is the handling cost at terminal i.

In order to build the related UND problem, the set of nodes V must be built from the set of terminals. Let each EOL terminal j be represented by a single node j in the set V, and let every breakbulk terminal i be split into two nodes, i_1 and i_2 in V. The set of arcs (or direct services) from the SLP problem can all be translated into the UND model by forcing any arc that ended at a breakbulk terminal i in the SLP model to end at the corresponding node i_1 in the UND model. Similarly, any arc that originated at a breakbulk terminal i in the SLP model, must originate at the corresponding node i_2 in the UND model. In addition to these arcs, there must be a single arc from i_1 to i_2 for all breakbulk terminals i which were split into two nodes. These additional arcs are referred to as the *split breakbulk arcs*, and they represent the actual handling at a terminal of any commodity routed along that arc.

Since our UND model requires a unit demand for each commodity, we must transform the commodity–dependent demands of the SLP problem to unit demands by scaling the costs for each commodity, as is done in Balakrishnan, Magnanti, and Wong[2] for their instances taken from the less–than–truckload, or LTL, industry. In other words, the value of c_{ij}^k for commodity $k \in K$ and for any arc $ij \in A$ which is not a split breakbulk arc is defined to be $transcost(ij) * frt(k)$. (Note that every arc $ij \in A$ which is not a split breakbulk arc corresponds to a unique arc in the SLP model. It is the transit cost of this original arc that

$transcost(ij)$ refers to.) The value of $c^k_{i_1 i_2}$ for commodity $k \in K$ and the split breakbulk arc $i_1 i_2$ obtained from terminal i is assigned the value of $handcost(i) * frt(k)$. Finally, the fixed cost F_{ij} of an arc $ij \in A$ which is not a split breakbulk arc is $minfreq(ij) * transcost(ij)$. For the split breakbulk arcs, $F_{ij} = 0$. Thus, we now have a related UND problem which we can solve heuristically using a known method.

A dual–ascent procedure for the UND

The method we use to solve the UND is essentially the dual–ascent procedure described in Balakrishnan, Magnanti and Wong [2]. This approach begins by first relaxing the binary constraints on the y variables in the primal UND problem (P). Then the dual is formed using the dual variables v^k_i for the flow conservation constraints and the dual variables w^k_{ij} for the second set of constraints (known as the forcing constraints). Thus, the dual UND problem (D) can be written as follows:

$$\begin{array}{lll}
\text{Maximize} & \sum_{k \in K} v^k_{D(k)} & \\
\text{subject to} & & \\
& v^k_j - v^k_i - w^k_{ij} \leq c^k_{ij} & ij \in A, k \in K \\
& \sum_{k \in K} w^k_{ij} \leq F_{ij} & ij \in A, k \in K \\
& w^k_{ij} \geq 0 & ij \in A, k \in K
\end{array}$$

Notice that when w is fixed, the remaining problem is the dual of a shortest paths problem. As a result, the solution v^k_i is the shortest path from $O(k)$ to i given that the cost on the arcs is the value of $c^k_{ij} + w^k_{ij}$. The dual-ascent algorithm takes advantage of this observation by initially fixing $w = 0$ (which is always feasible) and then finding the best values for v using a shortest paths algorithm. From this point, the values of w are iteratively increased with the v values being updated accordingly, until the objective of the dual can no longer be increased by simply increasing a component of w. This process in Balakrishnan, Magnanti, and Wong[2] is referred to as the unrestricted labeling method.

Within the labeling algorithm, we found that it is possible to eliminate several variables and thus improve efficiency. All of the variable eliminations are based on the differences between EOL and breakbulk terminals. Since an EOL terminal can never be an intermediate stop on a commodity's path, certain arcs do not make sense given a particular commodity. For example, if a commodity's origin and destination terminals are both breakbulks, then that commodity can never be routed along an arc that has an EOL terminal as one of its endpoints. Similarly, for a commodity with both EOL origin and destination terminals, no EOL terminal other than the commodity's origin or destination may appear in that commodities path. Thus, variables w^k_{ij} where the arc ij and the commodity k are incompatible may be eliminated. Also, variables v^k_i such that i is a terminal that can never appear on commodity k's path may also be eliminated. These observations were made in the paper by Balakrishnan, Magnanti, and Wong [2], but the implementation details were left to the programmer. These variable eliminations are done in our code, as well.

Once the labeling algorithm is completed, the information from the last iterate is used

to construct a primal UND solution. This method, though, does not necessarily yield a tree-based UND solution. To acquire a tree-based solution, we build a graph including only the arcs in the newly constructed primal solution. We then assign costs to the arcs based on their original commodity–independent transit costs and the handling cost at terminals for the split breakbulk arcs. A subgraph is then built for each terminal that is a destination for some commodity. This subgraph contains only the relevant arcs for the given destination. The direction of each of these arcs is reversed and the single–source shortest paths problem is solved with the destination as the source. This destination–rooted tree then provides the paths from every other node to the destination, given the arcs' original directions. This guarantees that the solution will be tree–based. We can then compute the corresponding primal UND objective value for the tree–based solution. To complete the load plan which must have a next transfer for every location/destination pair, we simply fill in unassigned pairs with the value from the historical load plan. Finally, we perform the arc–exclusion test from Balakrishnan, Magnanti, and Wong[2] which is based on the difference between the objective values of the best known primal and dual UND solutions. If any arcs are excluded, then we repeat the algorithm using the smaller network. The dual–ascent procedure terminates when no more arcs can be excluded using the arc–exclusion test.

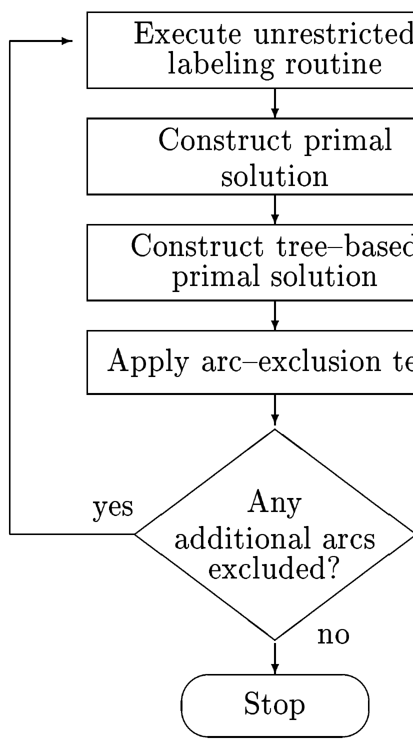

Figure 1: Flowchart for the dual–ascent procedure

3.3 Calculating the Objective

The UND solution provides a starting point upon which to improve. To evaluate the solution in terms of the SLP rather than the UND, we need to calculate the appropriate objective value. The SLP objective function is the sum of the transit cost of the vehicles used to ship the freight and the handling costs at each intermediate terminal through which the freight is routed. Given a load plan, determining the route taken by each commodity from its origin to its destination is straightforward. Thus, the handling costs are computed for each commodity by a direct examination of the intermediate handling terminals along this route.

Calculating the transit costs incurred shipping each commodity, however, cannot be determined so readily from the load plan. The load plan determines how much freight must be sent from one terminal to any other terminal, but it does not specify how to satisfy minimum frequency or flow conservation of trailers. Fortunately, given the load plan variables n_{ijd} for all $ij \in DIRS, \;\; d \in DEST$, our mathematical model decomposes into the following minimum cost network flow problem that deals with the minimum frequency and trailer balancing concerns in conjunction with satisfying freight distribution:

$$\begin{array}{lll} \text{Minimize} & \displaystyle\sum_{ij \in DIRS} TC_{ij} v_{ij} & \\ \text{subject to} & & \\ & \displaystyle\sum_{it \in DIRS} v_{it} - \sum_{ti \in DIRS} v_{ti} = 0 & t \in T \\ & v_{ij} \geq max \;\; (MF_{ij} n_{ijd} \,, \displaystyle\sum_{kd \in C} Q_{kd} f_{ijkd}) & ij \in DIRS, \;\; d \in DEST \end{array}$$

This minimum cost network flow problem specifies that each arc has a flow of trailers, represented by the v_{ij} variables, that is at least enough to transport the freight assigned to the direct and enough to satisfy minimum frequency if the direct is status in. This problem is solved using the CPLEX 5.0 callable library routine CPXhybnetopt [1]. From this optimization, we have the transit cost of the vehicles used to ship the freight which completes the calculation of the SLP objective function.

3.4 SLP Heuristics

In order to improve the level to which the UND approximates the SLP problem, we iteratively modify the cost vectors of the UND problem. The tree-based solutions constructed within the dual ascent algorithm are thus more likely to be good, feasible solutions to the SLP problem.

[1] CPLEX is a trademark of CPLEX Optimization, Inc.

In the historical solution for our primary data set each commodity with the exception of one is handled at no more than two terminals. Even though this is not a feasibility constraint for the SLP, this is an important requirement for most companies. In practice, handling a commodity at more than two handling points makes it difficult to meet service requirements under real–world conditions, even though the results from the mathematical model may appear to reduce costs by handling the commodities at several handling points. We implement an SLP heuristic that multiplies by a positive constant greater than one the per unit cost of every split breakbulk arc for a commodity that is handled more than two times. Recall from section 3.2 that split breakbulk arcs represent the actual handling cost at a terminal of any commodity routed along that arc. By not increasing the cost on every arc of a commodity and only increasing the cost on the split breakbulk arc, the algorithm is less likely to include as many of the more expensive split breakbulk arcs in its next solution.

For the SLP to be feasible, the path for each commodity may not exceed the service standards. Service is violated if the transit time for a commodity's path is greater than the commodity's service time plus a service tolerance parameter. The service tolerance is set to 60 minutes in our model. If service for a commodity is violated, the cost of every arc for a commodity is multiplied by 2. The cost on every arc is increased for this commodity so that the algorithm is more likely to choose a path with fewer arcs and not just an alternate path using as many arcs as before.

The minimum frequency requirement must also be met in order for the SLP to be feasible. The minimum frequency is the minimum number of trailers that must travel from the origin to the destination of a direct that is status in. Each time we compute the objective, we force the minimum frequency requirements to be satisfied. In our heuristics, if the number of loaded trailers on a direct is at least minimum frequency, the fixed cost for that direct is set to zero. Allowing the arcs that meet minimum frequency to be less expensive to use helps the UND better approximate the SLP.

The last requirement for the SLP to be feasible is that each terminal must not exceed its capacity. For our purposes, this is the least important feasibility requirement. If a terminal is over its capacity, the cost of every commodity for that terminal is multiplied by 2. Since the cost for the terminal that exceeds its capacity is increased for every commodity, fewer commodities are likely to be routed through this terminal, and thus the terminal is more likely to meet its capacity requirements.

An iterative loop is implemented to search for a "better" solution. We define a "better" solution by testing certain factors. We are most concerned with reducing the number of commodities that are handled at more than two intermediate breakbulks. We update our best solution to be the current solution if the total number of commodities over–handled in the current solution is less than the least amount of total commodities handled so far. Alternately, if handling is the same and there are less service violations than the HLP and a better objective value the best solution is reported. And finally, if the handling is the same and the service violations are less than the least service violations so far, or if the service is the same, but a better objective value is obtained, than the best solution can be updated. Since we are continuously multiplying the arc costs in the UND problem, causing them to

increase, there is a danger that these costs will attain values that are computationally too large. Therefore, we calculate the maximum number of iterations based on the maximum value any arc may be multiplied by during a single iteration. If the same solution is generated in successive iterations or the iteration limit is reached, the loop stops, and the best current solution is reported.

3.5 Add/Drop Heuristic

After finding solutions to the strategic load planning problem, we apply a simple add/drop heuristic to these solutions in an attempt to improve their overall quality. Our heuristic uses the original SLP commodity–independent cost structure for all of the arcs. We begin by considering only the arcs belonging to the inputted load plan. Given this subset of arcs, we can contruct a tree–based load plan as is done in the dual–ascent procedure. Using this load–plan, our heuristic examines each origin/destination pair as a potential add or drop candidate. If the direct corresponding to a pair is in the network and the freight shipped along it is less than the minimum frequency, then we consider it a drop candidate. On the other hand, if the direct corresponding to a pair is not in the network and the commodity associated with the origin/destination pair is handled more than twice or has at least minimum frequency amount of freight, then we consider it an add candidiate. Fixing a load plan after an arc has been added or dropped simply requires building a new tree–based solution based on the new subset of arcs. If the adding or dropping of the arc improves the solution, then the change is kept, otherwise the change is reversed back. An improved solution is one in which either

- the new objective value is lower than the best objective value found so far and neither handling or service violations are worse than those in the original tree–based solution, or
- the new objective value is qualitatively the same as the best objective value found so far and at least one of handling and service errors has improved and neither has worsened.

This process is repeated until an iteration without improvement is found.

4 Results

We performed our computational runs on a Sun Ultra 1 Model 140 with 192 megabytes of memory running Solaris 2.6. The overall structure of our algorithm is outlined in Figure 2. In order to fine–tune our algorithm to adapt to different data sets, we introduce parameters whose values we can vary prior to the start of any search for a feasible solution to the strategic load planning problem. Modifying these parameters is another way to make our mathematical model adhere to real–world considerations.

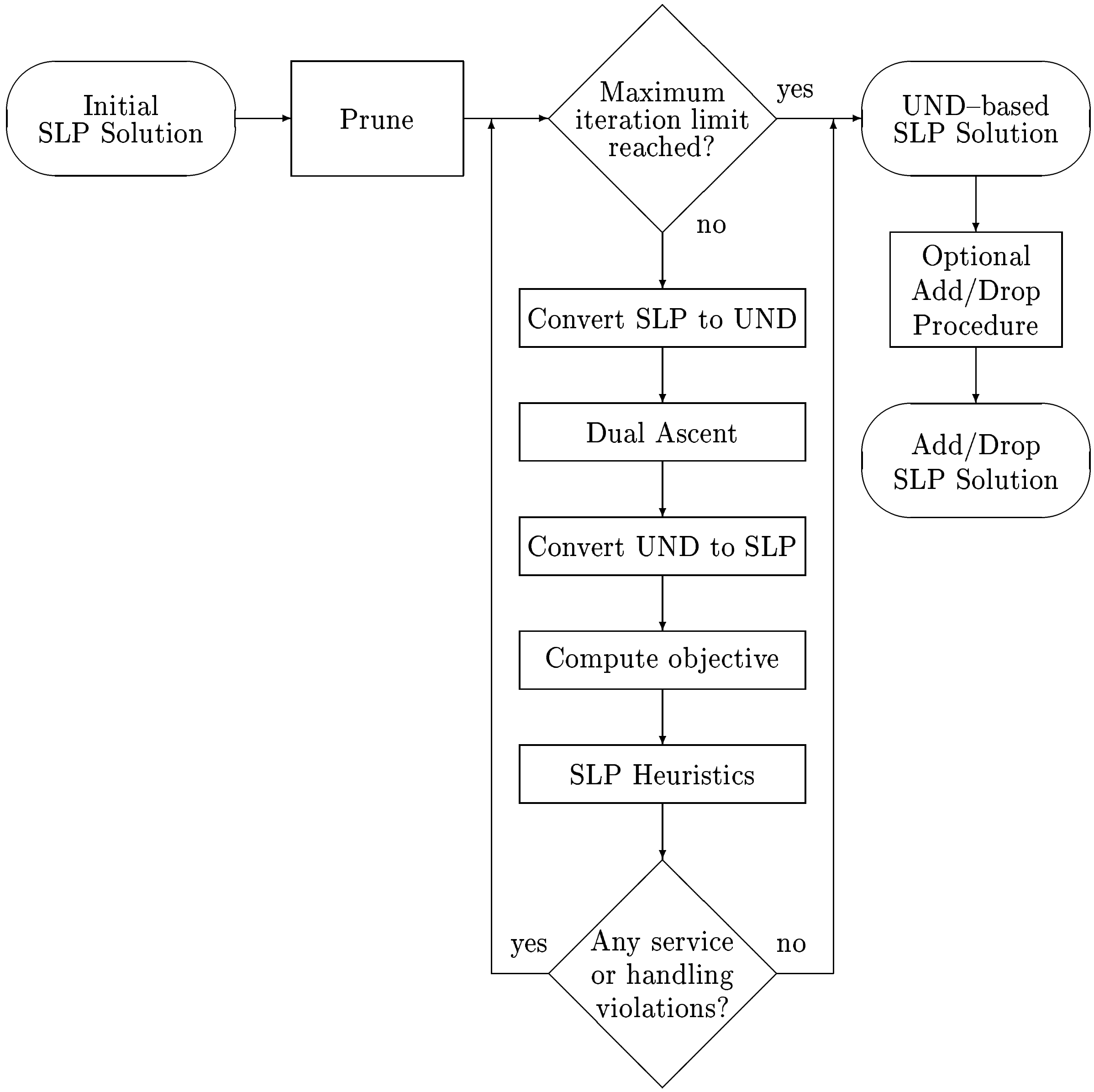

Figure 2: Flowchart for the complete algorithm

There are two parameters which influence the number of arcs which are pruned in the network. The low freight parameter refers to the pruning of EOL to EOL arcs. Since this procedure only considers removing a subset of arcs, varying the value of the parameter consequently has a limited effect. The pruning parameter used with the historical load summary,on the other hand, can potentially affect the entire list of arcs. Clearly, this technique is only relevant in instances where a load summary is provided.

Our original data assumes a fixed minimal number of hours for handling at any terminal. This throughput time at each terminal assumes that there is one main time each day (around midnight) when the majority of the freight to be transferred arrives and is handled in one very busy time window. Chances are high that a greater than two transfer freight path would fall outside this window for at least one of the transfer points. Thus, we use a parameter to

artificially inflate handling time and promote more directs and fewer handling points in the network.

Our original data also assumes 28,403 lbs/van and 14,202 lbs/pup when computing the number of trailer–volumes of freight for each commodity. The actual trailer capacities are closer to 35,000 lbs/van and 20,000 lbs/pup. If we increase the amount of freight that can be loaded onto each tractor, then we effectively increase handling costs relative to transit costs since handling costs are a function of the amount of freight, but transit costs are a function of the number of trailers used. A parameter is used to adjust the freight density of each type of trailer.

4.1 SEFL Data Set

Our primary data for this research comes from Southeastern Freight Lines (SEFL) headquartered in Lexington, SC. Four types of data were provided to us by SEFL. The first is terminal data, which includes the code for each terminal, the handling cost, the handling time, and the capacity. The capacity data, in this case, is less of a restriction and more of a suggestion as to how much freight is acceptable to transfer through each location. The SEFL network made available to us contains 48 terminals. The second set of data contains information about the possible direct services in the network. The data includes the origin terminal, the destination terminal, the cost per trailer, the transit time, and the minimum frequency for each direct. The third set of data describes the freight. We are given the origin terminal, destination terminal, trailer volumes per day, and the service standard advertised for each commodity. The trailer volumes per day is given as the number of trailers, possibly fractional, that the freight will fill. All freight is assumed to originate at 9:00 p.m. and is due by 8:00 a.m., local time. Therefore, the service times vary in small amounts depending on changes in time zones and in large amounts depending on the type of service (one, two, or three day service). The fourth and final type of data provided is an historical load summary which relates the number of times an arc is used by loaded trailers over a three month period. As is previously mentioned, this permits the use of an intelligent pruning technique.

In addition to this data, which defines the SLP problem, SEFL supplied an historical load plan, or HLP, representing what they considered to be a good solution in their network. The HLP is used inside our algorithm to maintain feasibility and is used outside the algorithm as a means of evaluating the quality of our solutions. In order to have a basis for comparison between the historical load plan and the various tests which we performed, we select one set of parameters corresponding to our original data set which we use to evaluate all of the solutions. The parameters correspond to (1) not changing the freight density (Frt. D. = 1.0); and (2) not increasing the handling time at a terminal beyond its initial setting (Handtime=0). In this framework, the HLP has no service violations, a very small number of handling violations (defined as a commodity being handled more than two times), and a small number of capacity violations. Actual values are not given here due to confidentiality constraints. All of the results are reported relative to these actual values, regardless of the individual test parameters.

4.1.1 SLP Heuristics

In general, all of the SLP heuristics worked as anticipated, but with varying degrees of success. The cost modifications implemented to improve the number of commodities meeting the service standard worked most efficiently. It worked well alone, as well as in conjunction with the other heuristics. As for the heuristic to handle commodities which were routed through more than two intermediate terminals, we found that this heuristic worked better when combined with the service heuristic. In other words, if only the handling heuristic is called at each iteration, we found that the number of commodities which were over–handled improved; but, if both the handling and the service heuristics were called at each iteration, the handling improved even faster (and with no negative consequences for the service heuristic). Therefore, we applied both heuristics at each iteration throughout our testing process.

In all of the preliminary testing, we discovered that the heuristic to reduce the number of capacity violations at terminals succeeded in reducing the capacity violations, but at great cost in terms of the objective function. Recall that this heuristic increases the cost of using a terminal that is over its capacity for every commodity, with no regard to the degree to which the constraint is violated. Rather than modifying the heuristic, we chose to simply disregard capacity violations in our model. This decision is based on two major factors. First, and most importantly, SEFL expressed some interest in a non–capacitated SLP model which could be used to better understand capacity needs at terminal locations. Additionally, the nature of the capacity data that was supplied reflects more of a soft constraint, than a strict requirement. Without more specific information, it should really be thought of as a guideline. Since we found that solutions obtained without considering capacity violations tended to have a comparable number of capacity violations to the HLP, we believe it likely that our solutions are within an acceptable range. Since comparison of solutions in terms of capacity violations is not essential to SEFL and difficult within the constraints of our data set, numerical results for capacity violations are not given. The only form of capacity restriction which we leave in the model is the definition of an EOL terminal. Thus, each terminal either has no capacity for handling freight (an EOL), or it has unlimited capacity (a breakbulk).

4.1.2 Parameter Variations

Three basic settings were modified to obtain our final results. The first setting concerns the number of arcs not in the HLP that are pruned. The EOL to EOL pruning parameter remained fixed at a reasonable value throughout all of the computation. The frequency parameter pertaining to the historical load summary, though, was used to prune varying proportions of those arcs not yet pruned and not in the HLP. As for the freight density parameter, we tested the values 1.0 (corresponding to no change in the original data), 1.1 and 1.2 which yield valid interpretations of trailer capacity. The third setting refers to the handling time at terminals. By adding a positive value to the through time at all terminals, we reduce the likelihood both of over–handling of freight and of service violations in the real

	Parameter Settings			UND–based Results			
Case	Prune	Frt.D.	Handtime	Handling	Service	Objective	Time (s)
SEFL.1	0.76	1.0	0	2.00	0	-0.006	355.34
SEFL.2	0.76	1.0	60	2.00	0	-0.001	370.95
SEFL.3	0.76	1.0	120	2.00	0	0.008	404.70
SEFL.4	0.76	1.1	0	1.00	0	-0.009	363.66
SEFL.5	0.76	1.1	60	1.00	0	-0.001	376.42
SEFL.6	0.76	1.1	120	1.00	0	0.010	412.75
SEFL.7	0.76	1.2	0	1.00	0	-0.013	466.33
SEFL.8	0.76	1.2	60	1.00	0	-0.007	356.60
SEFL.9	0.76	1.2	120	1.00	0	0.005	542.97
SEFL.10	0.90	1.0	0	1.00	0	-0.008	244.92
SEFL.11	0.90	1.0	60	2.00	0	-0.010	236.02
SEFL.12	0.90	1.0	120	2.00	0	-0.010	262.55
SEFL.13	0.91	1.1	0	2.00	0	-0.008	250.98
SEFL.14	0.91	1.1	60	2.00	0	-0.007	260.64
SEFL.15	0.91	1.1	120	2.00	0	-0.008	269.43
SEFL.16	0.91	1.2	0	0.00	0	-0.016	270.97
SEFL.17	0.91	1.2	60	0.00	0	-0.011	268.42
SEFL.18	0.91	1.2	120	0.00	0	-0.014	310.08
SEFL.19	0.93	1.0	0	2.00	0	-0.015	246.86
SEFL.20	0.93	1.0	60	2.00	0	-0.016	254.35
SEFL.21	0.93	1.0	120	2.00	0	-0.014	244.35
SEFL.22	0.93	1.1	0	5.00	0	-0.015	247.11
SEFL.23	0.93	1.1	60	2.00	0	-0.014	258.48
SEFL.24	0.93	1.1	120	2.00	0	-0.014	262.17
SEFL.25	0.93	1.2	0	3.00	0	-0.015	232.65
SEFL.26	0.93	1.2	60	2.00	0	-0.014	261.02
SEFL.27	0.93	1.2	120	2.00	0	-0.015	268.29

Table 1: This table reports results with varying settings for the prune, freight density and extra handling time parameters.

world. In our testing we set the extra handling time to 0, 60, or 120 minutes.

4.1.3 Numerical Results

In the following tables, the results are reported in relative terms, so as to not reveal confidential information about the SEFL network. The value in the "Prune" column refers to the proportion of arcs not in the HLP that are pruned. Thus, a 1.0 prune value indicates that only the arcs of the HLP were used to obtain the UND–based solution. The column marked "Frt. D." refers to the freight density setting and the column marked Handtime refers to the extra handling time added to all terminals. Handling errors are reported as the number

	Parameter Settings			UND–based Results			
Case	Prune	Frt.D.	Handtime	Handling	Service	Objective	Time (s)
SEFL.28	0.96	1.0	0	2.00	0	-0.014	225.92
SEFL.29	0.96	1.0	60	2.00	0	-0.016	234.23
SEFL.30	0.96	1.0	120	2.00	0	-0.014	239.75
SEFL.31	0.96	1.1	0	5.00	0	-0.013	209.55
SEFL.32	0.96	1.1	60	2.00	0	-0.014	236.31
SEFL.33	0.96	1.1	120	2.00	0	-0.014	243.78
SEFL.34	0.96	1.2	0	0.00	0	-0.012	215.10
SEFL.35	0.96	1.2	60	3.00	0	-0.017	238.46
SEFL.36	0.96	1.2	120	2.00	0	-0.012	266.39
SEFL.37	0.99	1.0	0	1.00	0	-0.017	192.13
SEFL.38	0.99	1.0	60	2.00	0	-0.018	185.95
SEFL.39	0.99	1.0	120	1.00	0	-0.018	179.15
SEFL.40	0.99	1.1	0	2.00	0	-0.015	196.14
SEFL.41	0.99	1.1	60	2.00	0	-0.017	190.68
SEFL.42	0.99	1.1	120	2.00	0	-0.017	286.39
SEFL.43	0.99	1.2	0	5.00	0	-0.017	194.70
SEFL.44	0.99	1.2	60	3.00	0	-0.019	190.12
SEFL.45	0.99	1.2	120	2.00	0	-0.017	220.99
SEFL.46	1.00	1.0	0	6.00	0	-0.018	178.25
SEFL.47	1.00	1.0	60	5.00	0	-0.018	189.99
SEFL.48	1.00	1.0	120	4.00	0	-0.017	170.07
SEFL.49	1.00	1.1	0	4.00	+	-0.018	150.81
SEFL.50	1.00	1.1	60	4.00	0	-0.019	191.81
SEFL.51	1.00	1.1	120	3.00	0	-0.017	230.11
SEFL.52	1.00	1.2	0	1.00	0	-0.020	181.97
SEFL.53	1.00	1.2	60	1.00	0	-0.021	192.57
SEFL.54	1.00	1.2	120	1.00	0	-0.017	251.99

Table 2: This table reports results with varying settings for the prune, freight density and extra handling time parameters.

of violations in the new solution minus the number of violations in the HLP divided by the number of violations in the HLP. Since the HLP contains zero service violations, a plus sign is used to indicate a positive number of violations and a zero is used to indicate no violations. The objective value refers to the new solution's obective value minus the HLP's objective value divided by the HLP objective value. Thus, a negative value in the column for handling errors or for objective values indicates an improvement over the HLP.

In addition to the above tests, we performed one additional series of runs involving a new data file for the terminal information. Rather than assume that all of the terminals have the same handling time, we were given data on the actual handling time for each terminal. We modified this data by forcing the through times to have a certain minimum value. This resulted in a set of terminals with greater through times which were more reflective of real

	Parameter Settings		UND–based Results			
Sedfile	Prune	Frt.D.	Hand. Errors	Serv. Errors	Objective	Time (s)
SEFL.55	0.93	1.0	-1.00	0	-0.014	262.52
SEFL.56	0.93	1.1	-1.00	0	-0.014	278.78
SEFL.57	0.93	1.2	-1.00	0	-0.012	268.01
SEFL.58	0.96	1.0	1.00	0	-0.015	226.20
SEFL.59	0.96	1.1	1.00	0	-0.015	228.28
SEFL.60	0.96	1.2	1.00	0	-0.010	281.33
SEFL.61	1.00	1.0	5.00	0	-0.016	170.32
SEFL.62	1.00	1.1	2.00	0	-0.016	201.90
SEFL.63	1.00	1.2	0.00	0	-0.016	238.55

Table 3: This table reports results obtained using a new terminal data file.

world conditions. With this new file, we ran tests similar to those in Tables 1 and 2, but without adding any extra handling time at the terminals. These solutions generally had fewer handling violations than their counterparts and can be found in Table 3.

Clearly the UND–based algorithm, prior to performing the add/drop heuristic, is capable of finding good solutions in our model. In every instance, the objective value is lower than the original HLP objective value. In fact, a 1% decrease (i. e. -0.01) in objective value can save the company several thousands of dollars a day. As for the feasibility issues, the service requirements are, for the most part, easily met. Considering that the initial number of handling violations is marginal, the number of handling violations in the solutions stays well within reason, and is even improved upon in certain instances. Some of the best results were obtained using the new file on terminal handling times. This seems to indicate that the better the data reflects the real problem, the better the overall performance of the UND–based algorithm. Pruning also has a large influence on the solution quality. Leaving only "good" arcs in the network for the UND not only decreases solution time, but also increases the improvement in solution quality. We suspect that the better the pruning technique, the better the resulting solution will be.

Finally, we applied the add/drop procedure to all of our UND–based solutions, as well as to the HLP. The results can be seen in Tables 4–6. All of these solutions are clearly superior to the original HLP, though, unfortunately, the add/drop on the HLP yields the overall best solution. It is not clear why this is the case. The results do indicate that the UND–based algorithm, in combination with the add/drop heuristic or standing alone, does do a credible job of building a strategic load plan. It is possible that there is something inherently better about SEFL's HLP that our model simply does not capture. This is difficult to evaluate, though. *[Real world evaluation here?]* It is also possible that our simplistic add/drop heuristic actually lessens the quality of our solutions by increasing the likelihood of service violations, given that we do not take into consideration any buffers on handling time at the terminals in the add/drop heuristic. In other words, though we maintain the same number of service violations while performing the add drop, the way in

Case	UND–based SLP solution				Add/Drop SLP solution			
	Handling	Service	Objective	Time (s)	Handling	Service	Objective	Time (s)
HLP	-	-	-	-	-1.00	0	-0.054	780.63
SEFL.1	2.00	0	-0.006	355.34	-1.00	0	-0.027	483.13
SEFL.2	2.00	0	-0.001	370.95	-1.00	0	-0.029	505.29
SEFL.3	2.00	0	0.008	404.70	-1.00	0	-0.039	748.34
SEFL.4	1.00	0	-0.009	363.66	-1.00	0	-0.019	343.24
SEFL.5	1.00	0	-0.001	376.42	-1.00	0	-0.029	502.62
SEFL.6	1.00	0	0.010	412.75	-1.00	0	-0.037	733.84
SEFL.7	1.00	0	-0.013	466.33	-1.00	0	-0.020	327.37
SEFL.8	1.00	0	-0.007	356.60	-1.00	0	-0.028	497.71
SEFL.9	1.00	0	0.005	542.97	-1.00	0	-0.037	636.01
SEFL.10	1.00	0	-0.008	244.92	-1.00	0	-0.029	338.55
SEFL.11	2.00	0	-0.010	236.02	-1.00	0	-0.034	360.63
SEFL.12	2.00	0	-0.010	262.55	-1.00	0	-0.040	402.06
SEFL.13	2.00	0	-0.008	250.98	-1.00	0	-0.027	335.80
SEFL.14	2.00	0	-0.007	260.64	-1.00	0	-0.033	362.50
SEFL.15	2.00	0	-0.008	269.43	-1.00	0	-0.038	527.28
SEFL.16	0.00	0	-0.016	270.97	-1.00	0	-0.033	320.52
SEFL.17	0.00	0	-0.011	268.42	-1.00	0	-0.034	400.37
SEFL.18	0.00	0	-0.014	310.08	-1.00	0	-0.041	856.02
SEFL.19	2.00	0	-0.015	246.86	-1.00	0	-0.030	303.18
SEFL.20	2.00	0	-0.016	254.35	-1.00	0	-0.034	448.24
SEFL.21	2.00	0	-0.014	244.35	-1.00	0	-0.040	524.33
SEFL.22	5.00	0	-0.015	247.11	-1.00	0	-0.029	313.55
SEFL.23	2.00	0	-0.014	258.48	-1.00	0	-0.033	432.47
SEFL.24	2.00	0	-0.014	262.17	-1.00	0	-0.039	517.95
SEFL.25	3.00	0	-0.015	232.65	-1.00	0	-0.029	308.27
SEFL.26	2.00	0	-0.014	261.02	-1.00	0	-0.033	429.30
SEFL.27	2.00	0	-0.015	268.29	-1.00	0	-0.039	506.21

Table 4: Compares the UND–based solution to the solution obtained after performing the add/drop heuristic.

which we calculate a service violation is always using the original terminal data with the minimal, fixed through times. As discussed previously, it is not clear that this is the best way in which to model the SLP problem. Since it is, however, the setting in which we were given the HLP, it is the one to which we adhere. Also note that the entire algorithm is easily completed in under 20 minutes from start to finish.

Case	UND–based SLP solution				Add/Drop SLP solution			
	Handling	Service	Objective	Time (s)	Handling	Service	Objective	Time (s)
HLP	-	-	-	-	-1.00	0	-0.054	780.63
SEFL.28	2.00	0	-0.014	225.92	-1.00	0	-0.031	318.00
SEFL.29	2.00	0	-0.016	234.23	-1.00	0	-0.034	333.80
SEFL.30	2.00	0	-0.014	239.75	-1.00	0	-0.038	496.61
SEFL.31	5.00	0	-0.013	209.55	-1.00	0	-0.028	322.79
SEFL.32	2.00	0	-0.014	236.31	-1.00	0	-0.032	322.67
SEFL.33	2.00	0	-0.014	243.78	-1.00	0	-0.037	483.41
SEFL.34	0.00	0	-0.012	215.10	-1.00	0	-0.027	331.57
SEFL.35	3.00	0	-0.017	238.46	-1.00	0	-0.033	439.50
SEFL.36	2.00	0	-0.012	266.39	-1.00	0	-0.037	492.88
SEFL.37	1.00	0	-0.017	192.13	-1.00	0	-0.032	301.50
SEFL.38	2.00	0	-0.018	185.95	-1.00	0	-0.036	311.93
SEFL.39	1.00	0	-0.018	179.15	-1.00	0	-0.038	376.25
SEFL.40	2.00	0	-0.015	196.14	-1.00	0	-0.029	287.14
SEFL.41	2.00	0	-0.017	190.68	-1.00	0	-0.033	306.35
SEFL.42	2.00	0	-0.017	286.39	-1.00	0	-0.037	480.32
SEFL.43	5.00	0	-0.017	194.70	-1.00	0	-0.029	290.54
SEFL.44	3.00	0	-0.019	190.12	-1.00	0	-0.035	422.16
SEFL.45	2.00	0	-0.017	220.99	-1.00	0	-0.037	484.26
SEFL.46	6.00	0	-0.018	178.25	-1.00	0	-0.018	295.55
SEFL.47	5.00	0	-0.018	189.99	-1.00	0	-0.026	328.02
SEFL.48	4.00	0	-0.017	170.07	-1.00	0	-0.028	383.45
SEFL.49	4.00	+	-0.018	150.81	-1.00	+	-0.022	294.49
SEFL.50	4.00	0	-0.019	191.81	-1.00	0	-0.025	331.74
SEFL.51	3.00	0	-0.017	230.11	-1.00	0	-0.030	344.90
SEFL.52	1.00	0	-0.020	181.97	-1.00	0	-0.022	279.94
SEFL.53	1.00	0	-0.021	192.57	-1.00	0	-0.027	326.44
SEFL.54	1.00	0	-0.017	251.99	-1.00	0	-0.034	453.39

Table 5: Compares the UND–based solution to the solution obtained after performing the add/drop heuristic.

4.2 Averitt Data set

Since the code was initially tailored to the SEFL data set, we obtained a second set of data to test the effectiveness of our algorithm. This data comes from the an alternate carrier called Averitt. The network they provided contains 78 terminals, rather than 48. Essentially they made available to us the same type of data as did SEFL. The one exception is the lack of an historical load summary. Though very useful as a pruning tool, this data is not essential to our method. Averitt also supplied an historical load plan, which we will continue to refer to as the HLP. In this case, the HLP contained several handling and service violations. Solutions are evaluated relative to the HLP solution, as in the SEFL section. Since the

Case	UND–based SLP solution				Add/Drop SLP solution			
	Handling	Service	Objective	Time (s)	Handling	Service	Objective	Time (s)
HLP	-	-	-	-	-1.00	0	-0.054	780.63
SEFL.55	-1.00	0	-0.014	262.52	-1.00	0	-0.044	427.77
SEFL.56	-1.00	0	-0.014	278.78	-1.00	0	-0.044	563.49
SEFL.57	-1.00	0	-0.012	268.01	-1.00	0	-0.044	546.68
SEFL.58	1.00	0	-0.015	226.20	-1.00	0	-0.039	538.33
SEFL.59	1.00	0	-0.015	228.28	-1.00	0	-0.038	489.90
SEFL.60	1.00	0	-0.010	281.33	-1.00	0	-0.038	505.36
SEFL.61	5.00	0	-0.016	170.32	-1.00	0	-0.026	387.80
SEFL.62	2.00	0	-0.016	201.90	-1.00	0	-0.031	353.20
SEFL.63	0.00	0	-0.016	238.55	-1.00	0	-0.035	505.01

Table 6: Compares the UND–based solution to the solution obtained after performing the add/drop heuristic.

number of service violations in the HLP for the Averitt data set is nonzero, the relative calculation of service errors is done in the same manner as the handling errors. For the same reasons as previously discussed, we continue to disregard possible capacity violations in the model.

Since the Averitt data does not include a load summary, the pruning techniques available to us are more limited. The technique in which low freight EOL to EOL arcs not in the HLP are eliminated is now the only flexible pruning method we have. In the first set of tests performed in Table 7, this low freight parameter is set to a reasonably low value. In the second set of tests, the low freight parameter is increased so that all EOL to EOL arcs not in the HLP are eliminated. Finally, in the third set of runs all arcs not in the HLP are pruned. The different settings for freight density and extra handling time remain the same.

Table 7 displays the results obtained by running the UND–based algorithm on the Averitt data set. The larger network resulted in a considerable increase in time. Creating a strategic load plan is not a task that is likely to be performed frequently. Thus, the increase in time is not unreasonable, especially since the solutions obtained are, overall, much better than the HLP solution. It is particularly interesting to note that the solutions obtained using only the HLP arcs in the network have the greatest improvement in objective function value, and the worst performance in terms of handling and service violations. We infer that if a better pruning technique were available to us – one in which we could prune approximately 90% of the arcs not in the HLP – we could obtain even better solutions than those in Table 7. As the results stand, the solutions are quite good. In this problem instance, a decrease in the objective of 1% also translates into a couple of thousands of dollars a day.

Due to the increased time involved in solving problems in this data set, we chose to only run the add/drop portion of our algorithm on three UND–based solutions and the HLP. As can be seen in Table 8, two out of three of our solutions are clearly superior to the HLP

	Parameter Settings			UND–based Results			
Case	Prune	Frt.D.	Handling	Hand. Errors	Serv. Errors	Objective	Time (s)
AVE.1	0.58	1.0	0	-1.00	-0.23	-0.093	4678.31
AVE.2	0.58	1.0	60	-1.00	-0.23	-0.067	4248.93
AVE.3	0.58	1.0	120	-1.00	-0.15	-0.001	4471.12
AVE.4	0.59	1.1	0	-1.00	-0.19	-0.085	5221.64
AVE.5	0.59	1.1	60	-1.00	-0.27	-0.060	4967.60
AVE.6	0.59	1.1	120	-1.00	-0.10	-0.006	4584.90
AVE.7	0.61	1.2	0	-1.00	-0.27	-0.088	4718.58
AVE.8	0.61	1.2	60	-1.00	-0.19	-0.070	5013.90
AVE.9	0.61	1.2	120	-0.73	-0.10	-0.009	4662.14
AVE.10	0.72	1.0	0	-1.00	-0.23	-0.093	3631.35
AVE.11	0.72	1.0	60	-1.00	-0.23	-0.067	3290.05
AVE.12	0.72	1.0	120	-1.00	-0.15	-0.001	3458.89
AVE.13	0.72	1.1	0	-1.00	-0.19	-0.085	4113.02
AVE.14	0.72	1.1	60	-1.00	-0.27	-0.060	3920.00
AVE.15	0.72	1.1	120	-1.00	-0.10	-0.006	3583.44
AVE.16	0.72	1.2	0	-1.00	-0.27	-0.088	3786.37
AVE.17	0.72	1.2	60	-1.00	-0.19	-0.070	4020.06
AVE.18	0.72	1.2	120	-0.73	-0.10	-0.009	3737.60
AVE.19	1.00	1.0	0	1.64	0.17	-0.142	747.09
AVE.20	1.00	1.0	60	1.64	0.19	-0.134	1034.41
AVE.21	1.00	1.0	120	1.64	0.19	-0.127	1001.86
AVE.22	1.00	1.1	0	1.55	0.17	-0.141	760.95
AVE.23	1.00	1.1	60	1.55	0.19	-0.134	1059.33
AVE.24	1.00	1.1	120	1.55	0.19	-0.127	1012.96
AVE.25	1.00	1.2	0	1.36	0.17	-0.139	778.31
AVE.26	1.00	1.2	60	1.36	0.19	-0.131	1072.61
AVE.27	1.00	1.2	120	1.36	0.21	-0.125	1033.71

Table 7: UND–based results for the Averitt data set.

	UND–based SLP solution				Add/Drop SLP solution			
Case	Handling	Service	Objective	Time (s)	Handling	Service	Objective	Time (s)
HLP	-	-	-	-	0.18	0.08	-0.084	11680.50
AVE.7	-1.00	-0.27	-0.088	4718.58	-1.00	-0.31	-0.146	7916.99
AVE.14	-1.00	-0.27	-0.060	3920.00	-1.00	-0.35	-0.147	8827.30
AVE.22	1.55	0.17	-0.141	760.95	1.55	0.17	-0.142	7909.60

Table 8: Add/drop results for the Averitt data set.

and the add/drop on the HLP. Though it may look odd, it is quite true that the number of handling and service errors increased in the solution to the add/drop on the HLP. This is

due to the fact that our add/drop heuristic is based on finding shortest paths in terms of cost, not time. So, the very first step of the add/drop heuristic is to find a solution using the shortest paths in terms of cost along only the arcs of the inputted SLP solution. Generally this solution is quite similar to the inputted load plan, though it is not always as close as we would desire. Our add/drop heuristic can not consider both cost and time tradeoffs as well as the UND–based algorithm; and in this case, the handling and service errors increased because of this. It is interesting to note, as well, that the time it takes to run our complete algorithm is comparable to the time it takes to run just the add/drop on the HLP. Thus, in approximately the same amount of time, our method found a better solution.

4.3 Daiichi Data set

Our final set of data was provided by Daiichi. This network containing 92 terminals is the largest of our three data sets. Due to its size, computation on this data set had to be performed on a larger machine. All of the computation on the Daiichi data set was performed on a Sun MicroSystems i86pc with 1024 megabytes of memory.

	Parameter Settings			Results			
Case	Prune	Frt.D.	Handtime	Handling	Service	Objective	Time (s)
DAIICHI.1	1.00	1.0	0	-0.87	0.10	-0.180	3554.78
DAIICHI.2	1.00	1.0	60	-0.88	0.10	-0.169	3520.67
DAIICHI.3	1.00	1.0	120	-0.88	0.10	-0.161	3477.18
DAIICHI.4	1.00	1.1	0	-0.89	0.10	-0.180	3621.57
DAIICHI.5	1.00	1.1	60	-0.90	0.10	-0.166	3577.00
DAIICHI.6	1.00	1.1	120	-0.90	0.10	-0.157	3530.64
DAIICHI.7	1.00	1.2	0	-0.91	0.09	-0.179	3708.29
DAIICHI.8	1.00	1.2	60	-0.92	0.10	-0.166	3650.26
DAIICHI.9	1.00	1.2	120	-0.92	0.10	-0.156	3599.61
DAIICHI.10	0.02	1.2	0	-0.99	-0.74	0.580	26891.51
DAIICHI.11	0.02	1.2	60	-1.00	-0.74	0.665	26334.59
DAIICHI.12	0.02	1.2	120	-1.00	-0.76	0.759	25780.46

Table 9: UND–based results for the Daiichi data set.

As in the Averitt case, the Daiichi data did not include an historical load summary. The HLP for the Daiichi data contains many more handling and service violations than the previous HLP solutions, leaving more room for improvement. In this case we either prune all of the arcs not in the HLP, or all of the EOL to EOL arcs not in the HLP. The significant difference in the number of arcs in the network is apparent from Table 9. Using our low freight pruning technique to its full potential, still only eliminates 2% of the prunable arcs. Fewer tests were done in conjunction with the EOL pruning technique since the considerable number of arcs prevented finding good overall solutions. The results from the Daiichi data clearly indicate the importance of an intelligent pruning technique. While the solutions

obtained by considering only the HLP arcs are certainly superior to the HLP solution, it is likely that a better pruning technique based on more data from the company would provide even better quality solutions.

Case	UND–based SLP solution				Add/Drop SLP solution			
	Handling	Service	Objective	Time (s)	Handling	Service	Objective	Time (s)
HLP	-	-	-	-	-0.99	-0.04	-0.288	31647.09
DAIICHI.7	-0.91	0.09	-0.179	3708.29	-0.99	-0.05	-0.281	36504.32
DAIICHI.8	-0.92	0.10	-0.166	3650.26	-0.99	-0.05	-0.281	37357.59
DAIICHI.9	-0.92	0.10	-0.156	3599.61	-0.99	-0.05	-0.283	37089.78

Table 10: Add/drop results for the Daiichi data set.

Table 10 indicates the comparable solution quality of the add/drop solutions for both UND–based solutions and the HLP. Unfortunately, though, the add/drop heuristic on a network of this size takes 8 to 10 hours. Since, the UND–based algorithm only requires approximately one hour to return an improved solution, it is likely that this method, combined with an intelligent pruning technique, is a better alternative for a company whose network is as large as this one.

5 Conclusions and Future Work

Our complete algorithm, consisting of a modified uncapacitated network design method and an add/drop procedure, found good solutions to the strategic load planning problem for the data sets tested. These solutions were found in a reasonable amount of time and were qualitatively on par with the historical load plans implemented by the carriers. The UND-based portion of the algorithm consistently improved upon the initial solution given, as did the add/drop procedure. Surprisingly, however, when the add/drop procedure is applied to the historical load plans from the carriers SEFL and Daiichi, the solutions generated are better than the solutions generated from add/drop when applied to the UND-based solutions. This result holds regardless of whether or not the UND-based solutions inputted into the add/drop procedure were better than the corresponding historical load plans. We do not believe that this result reflects negatively on our algorithm, but that it strongly confirms the quality of time-tested solutions used by the carriers.

In terms of future work, we believe that better pruning techniques would improve the efficacy and applicability of our algorithm. Currently, we rely on historical data provided by the carriers for our most effective pruning heuristic. However, we would like to able to ascertain which directs are reasonable or unreasonable based upon real-world considerations that apply generally to all carriers.

The terminal capacities were ignored in our algorithm because the company from which our primary data came was interested in deciding whether or not to add to an existing

terminal or build a new one. Ignoring terminal capacities becomes a problem if we are approached by a carrier for whom this is a hard constraint. We believe that the current but unused capacity heuristic can be improved by basing our cost multiplier on the degree of capacity violation as opposed to using a constant multiplier.

Lastly, we did not fully address the issue of actively improving minimum frequency. Currently, if the total trailer-loads of freight plus the number of empty trailers on a direct meets the minimum frequency requirement, then we assume that we can safely reduce the fixed cost associated with this direct to zero. On the other hand, if the amount of freight and the number of empty trailers falls far below the minimum frequency on an edge, then we do not change the fixed costs for the edge. For edges that fall between these two extremes the fixed costs could be modified as a function of the difference between the total number of trailers assigned (loaded and empty) and the minimum frequency.

6 Acknowledgements

We would like to thank the carrier Southeastern Freight Lines, and in particular Phil Teague and Braxton Vick for making this data available to us. Without their help, this work would not have been possible.

References

[1] Anantaram Balakrishnan, Thomas L. Magnanti, and Prakash Mirchandani. Network Design. Annotated Bibliographies in Combinatorial Optimization: 311-334, 1997.

[2] A. Balakrishnan, T. L. Magnanti and R.T. Wong. A Dual-Ascent Procedure for Large-Scale Uncapacitated Network Design. Operations Research 37(5): 716-740, 1989.

[3] John W. Billheimer and P. Gray. Network Design with Fixed and Variable Cost Elements. Trans. Sci. 7: 49-74, 1973.

[4] John W. Braklow, William W. Graham, Stephen M. Hassler, Ken E. Peck, Warren B. Powell. Interactive Optimization Improves Service and Performance for Yellow Freight System. Interfaces 22: 147-172, 1992.

[5] *Using the CPLEX Callable Library.* CPLEX Optimization, Incline Village, NV, 1989-1995.

[6] R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language For Mathematical Programming.* The Scientific Press, San Francisco, CA, 1993.

[7] Michaël Gryseels, Fabrice Poppe, and Piet Demeester. A Fast Combinatorial Heuristic for Capacity Installation Problems with Multiple Facility Types. Department of Information Technology, University of Ghent, St. Pietersnieuwstraat 41, 9000 Ghent (Belgium), 1997.

[8] Kaj Holmberg and Di Yuan. Lagrangean Heuristic Based Branch-and-Bound Approach for the Capacitated Network Design Problem. Department of Mathematics, Linköping Institute of Technology, S-581 83 Linköping, Sweden, 1996.

[9] Introduction to LTL Trucking. SABRE, Inc., 22 Third Avenue, Burlington, Massachusetts 01803, 1997.

[10] Bruce W. Lamar, Yosef Sheffi, Warren B. Powell. A Capacity Improvement Lower Bound for Fixed Charge Network Design Problems. Operations Research 38(4): 704-710, 1990.

[11] Marc Los and Christian Lardinois. Combinatorial Programming, Statistical Optimization and the Optimal Transportation Network Problem. Transportation Research-B 16B(2): 89-124, 1982.

[12] T. L. Magnanti, P. Mireault, and R. T. Wong. Tailoring Benders Decomposition for Uncapacitated Network Design. Mathematical Programming Study 26: 112-154, 1986.

[13] W. B. Powell and Y. Sheffi. Design and Implementation of an Interactive Optimization System for Network Design in the Motor Carrier Industry. Operations Research 37(1): 12-29, 1989.