

Routing in Point-to-Point Delivery  
Systems

by

Janny M.Y. Leung, Thomas L. Magnanti,  
and  
Vijay Singhal

OR 174-88

January 1988

\*\*This paper is also printed as a Working Paper at the Yale School of  
Organization and Management, Series B, No. 103.

# Routing in Point-to-Point Delivery Systems

Janny M. Y. Leung

*School of Organization & Management, Yale University*

Thomas L. Magnanti

*Sloan School of Management, MIT*

Vijay Singhal

*Resource Management Systems*

January 21, 1988

## Abstract

We develop an optimization-based approach for a point-to-point route planning problem that arises in many large scale delivery systems (for example, less-than-truckload freight, rail, mail and package delivery, communications). In these settings, a firm which must ship goods between many origin and destination pairs on a network needs to specify a route for each origin-destination pair so as to minimize transportation costs and/or transit times. Typically, the cost structure is very complicated. The approach discussed in this paper exploits the structure of the problem to decompose it into two smaller subproblems, each amenable to solution by a combination of optimization and heuristic techniques. One subproblem is an 'assignment' problem with capacity constraints. The other subproblem is a mixed-integer multicommodity flow problem. We propose solution methods based on Lagrangian relaxation for each subproblem. Computational results with these methods and with a heuristic procedure for the multicommodity flow problem on a problem met in practice are encouraging and suggest that mathematical programming methods can be successfully applied to large-scale problems in delivery systems planning and other problems in logistical system design.

In many large scale delivery systems, a firm must ship goods between many origin and destination pairs of nodes on a network. Delivery route planning attempts to specify a route for each origin-destination pair so as to minimize transportation costs and/or transit times. The complication in many delivery systems is that the costs are nonlinear, nonseparable by origin-destination pairs, and usually include fixed charges.

This type of point-to-point delivery route planning problem arises in numerous applications settings, for example, package delivery, telephone circuit switching, mail delivery and rail freight shipping. As the service sector's share of the GNP increases, these application contexts will continue to grow in scope and importance. For example, from 1975 to 1981, total outlays for freight transportation doubled from \$ 115.5 billion to \$234.7 billion (*Transportation in America*, 1983). In addition, the globalization of many industries adds new demands for system integration of transportation and communication, and for more rapid and reliable point-to-point delivery services. As but one indication of the changes taking place, international air freight increased 16.2 percent from 1985 to 1986 (*U.S. Industrial Outlook*, 1987).

Effective planning in these settings typically poses significant challenges to managers and analysts because of the enormous scale of the underlying routing networks. For example, the telephone network in the United States connects over a hundred million subscribers, and the postal service delivers over a billion pieces of mail per year. In the real-world application that motivates this research, a cost savings of one percent translates into profits of millions of dollars a year.

Despite the demonstrated importance of these problems and their associated high payoffs, most problems of practical interest have yet to be solved successfully by mathematical programming methods. This gap between practice and relevant planning methods serves as a testimony to the difficulty of these problems.

In this paper, we present a model and solution methods for such delivery systems. The model is a complex large-scale non-linear mixed integer program. Our solution approach decomposes the problem into two smaller

subproblems:

1. a capacitated “assignment” problem that models the assignment of origin-destination pairs to pairs of intermediate distribution centers, and
2. a mixed-integer multicommodity flow routing problem that specifies routes connecting the intermediate distribution centers.

We propose to solve each subproblem by integer programming decomposition (Lagrangian relaxation) and heuristic methods.

This research originates from a project we conducted with a large-scale transportation carrier. The algorithms we developed form part of a decision support system used for operations planning. As reported in Section 7, our computational experience with this application indicates cost savings of 5 percent.

## 1 The Delivery Route Planning Problem

We first discuss briefly some of the features of the delivery systems we are trying to model.

The delivery network contains nodes of two types — terminal nodes and distribution centers (see Figure 1). All traffic originates and is destined for terminal nodes. Distribution centers are intermediate transshipment nodes at which the shipper may consolidate or disaggregate (break down) goods for distribution. The required volume to be shipped for each origin-destination pair is specified a priori, and the objective is to specify a route for each origin-destination pair that minimizes total system costs. There are two cost components — shipping costs on each link of the network and processing costs at each node.

Note that we are modeling shipments between origin-destination pairs and not the collection/distribution of goods from/to customers. For ex-

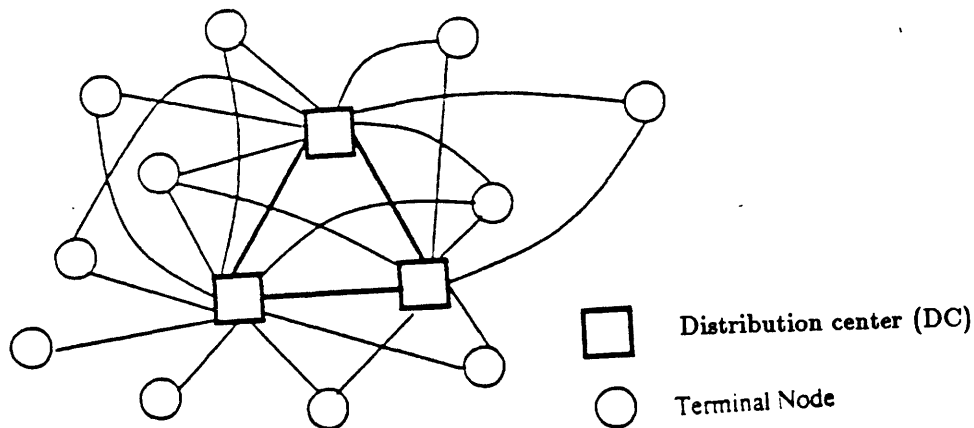


Figure 1: A sample delivery network

ample, in mail delivery the terminals will correspond to local post offices, and the distribution centers will correspond to major sorting facilities. We are not, therefore, modeling the local pickup and delivery of mail from end customers (which typically is a vehicle routing problem).

The goods are shipped on a link in trailerloads (trainloads or communication packets in other settings), and so the shipping cost on a link is a step function of the amount of goods sent on a link. Figure 2 shows an example of the link shipping costs. Notice that the 'steps' might not be even since the shipper might use several possible trailer sizes (e.g., 24-foot trailers, 48-foot trailers, double trailers). In practice, even the capacity for each trailer type is not fixed because the items are not of uniform size.

The processing at a distribution center involves the following operations:

1. unloading arriving trailers,
2. sorting the unloaded items by destination, and
3. reloading the sorted items onto trailers to be dispatched to other distribution centers or to terminals.

The processing cost at the distribution centers as a function of the volume to be sorted has the typical nonlinear functional form shown in Figure 3.

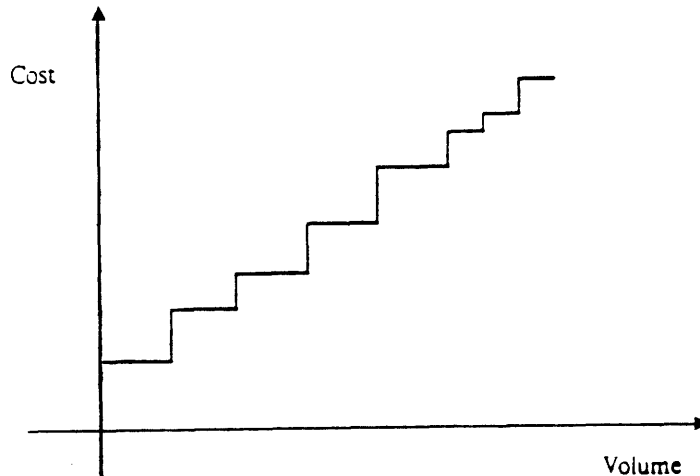


Figure 2: Graph of Link Shipping Cost as Function of Volume.

Each distribution center (henceforth called a DC node) has a nominal capacity, beyond which the costs increase rapidly due to congestion, as well as a minimum level of efficiency, usually about 50% of capacity. When the DC node operates within these limits, the processing cost per item is approximately constant. In the application context that motivated this study, and in many similar applications, the location of distribution centers is not a major concern; the firm would occasionally contemplate a very few additions or deletions of distribution centers (at a very high cost). It would analyze such changes by considering a very few alternate scenarios, using the type of route planning application considered in this paper to evaluate the possible scenarios. Consequently, in contrast with the plant/warehouse location literature, the problem we are investigating need not encompass fixed costs at the distribution centers.

In practice, the cost and route structure is even more complicated. Driver work rules might limit the distance between terminals and DC's. At times, the shippers can avoid the processing at a DC node by dispatching a full trailerload directly on to the next DC node without sorting. (In these instances, the DC might serve merely as a drop off point for an exchange of drivers.) Moreover, operationally it is often impossible to allow multiple routes for each origin-destination pair. Single routes simplify sys-

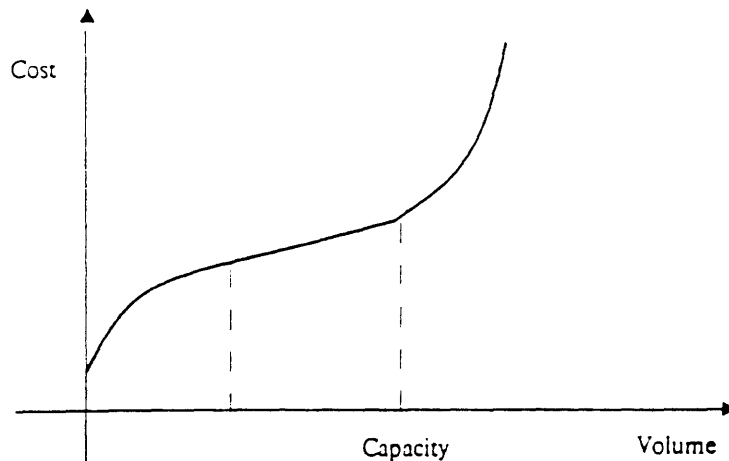


Figure 3: Distribution Center Processing Cost as Function of Volume.

tem control as well as operations at the DC where operating staff typically sort goods only by final destination. In addition, flows must satisfy service requirements, which guarantee delivery within a certain time frame. These service requirements prevent goods from being sent on a very indirect route to save on transportation and processing costs.

Other issues to be considered include the balance of flow of trailers and the number of destinations that a DC node can handle. Because these considerations are of secondary importance for our application context, we do not include them in the mathematical models we will present.

## 1.1 Size of the Problem

The test data from the real-world problem that motivates this study has a network with approximately 250 nodes (of which about 40 are distribution centers), 2000 links and 8000 origin-destination pairs. As a device to model service requirements (timing features) adequately, we let each node of the network represent an activity at a particular location at a specific time of day; thus the number of nodes in the network representation is often four or five times the number of physical locations in the system.



For this network, a mixed integer multicommodity flow formulation of the route planning problem (see Section 5) contains sixteen million zero-one variables, which is clearly beyond the capacity of current state-of-the-art general integer programming codes. The route planning problem cannot be solved as a monolithic integer program and we must find a way to reduce it to a manageable size. We also feel that a modular approach to solving the problem might be attractive practically since it allows more flexibility in adjusting specific portions of the model as data and problem ingredients change over time.

## 2 Related Research

The operations research and transportation service communities have conducted considerable research on distribution systems design and route planning; we will not attempt to review the literature in detail. Comprehensive surveys by Eilon, Watson-Gandy and Christofides [1971], Geoffrion [1975], Magnanti [1981], Bodin, Golden, Assad and Ball [1983], Magnanti and Wong [1984] and Dejax and Crainic [1987] summarize much of this literature. The problem studied in this paper does not fit the usual vehicle routing framework because goods are not carried from their origin to their destination on a single vehicle. Moreover, because of the step increments in cost on the links, the cost structure is complicated and differs from the usual plant/warehouse location literature, which focuses on location and sizing of the distribution centers (i.e., the fixed cost depicted in Figure 3). Similar problems that involve both assignment and routing with discontinuous link cost functions often lead to large mixed integer programming models. Some examples are the less-than-truckload loading problem (see Powell and Sheffi [1983]), the train blocking problem (see Bodin, Golden, Schuster and Romig [1980] and Assad [1980]), and the warehouse location-routing problem (see Perl and Daskin [1985]).

Fisher and Jaikumar [1981] studied a vehicle routing problem and used a “dual-ascent” solution strategy similar in spirit to the one we propose in Section 5. In studying the less-than-truckload problem, Balakrishnan and

Graves [1985] applied a dual ascent approach. Lamar, Powell and Sheffi [1984] also investigated the LTL loading problem, using ideas from fixed cost network design (see also Balakrishnan, Magnanti and Wong [1987]).

The modeling approach adopted in this paper addresses tactical planning issues: route selection, operating levels of the distribution centers, and vehicle loading. As we indicate in our brief discussion of implementation issues (Section 7.4 and 8), the model can also be used to address a variety of “what if” and strategic planning issues (e.g. location of distribution centers and terminals). Other, more aggregate, planning tools might also prove useful for addressing these system design issues: for example, the macroscopic approach of Daganzo [1987] or of Burns, Hall, Blumenfeld and Daganzo [1985]. Indeed, a macroscopic approach can sometimes lead to surprising and insightful results: for example, in many logistics location problems, such as locating the terminals in our problem setting, facilities should be placed as centers of hexagons in a hexagonal partitioning of the service region (see Fejes Toth [1953], Papadimitriou [1981] or Haimovich and Magnanti [1987]). These results seem more than academic: in the past, our industrial collaborators has used this design strategy to locate its terminals.

### **3 A Decomposition of the Routing Problem**

We propose a decomposition approach to the route planning problem which separates it into two linked subproblems. Each subproblem is solved ‘approximately’ to generate a ‘good’ solution to the overall problem.

Consider the routing decision for each origin-destination pair in two steps:

1. the assignment of each origin-destination pair to a (first DC)-(last DC) pair,
2. the choice of route from the first DC node to the last DC node.

Separating the routing decision into these two steps decomposes the route planning problem into two subproblems. The first sub-problem considers the assignment of a first and last DC node on the route for every origin-destination terminal node pair. After all such assignments have been made, the second subproblem seeks a minimum cost routing of the aggregated flow of goods among the DC nodes. This decomposition forces all items with the same first and last DC nodes to be shipped via the same intermediate nodes irrespective of their origin and destination terminal nodes. This property of the route plan greatly simplifies the processing task at the DC nodes, and is a very desirable feature from an operational standpoint.

There are several other advantages in adopting this decomposition approach. Focusing the routing problem on the DC network rather than the complete network significantly reduces the size of the routing problem. Not only does this approach considerably reduce the size of the network, it also dramatically decreases the number of origin-destination pairs because of volume aggregation. In effect, this approach collapses the original network into an aggregate network consisting only of the DC nodes. With this reduction in problem size, we can formulate a more detailed and accurate model of the routing problem and investigate optimization-based solution methods or heuristic procedures that would be prohibited by the size of the complete problem.

A second advantage of the decomposition approach is that it decouples the major decisions of the problem and simplifies the problem structure. Effectively, the routing problem acts as a costing procedure for pricing out a given assignment. From this perspective, we can view the complete problem as that of finding the terminal-to-DC assignment with the minimum total cost.

However, the two problems are linked and cannot be optimized separately. In solving the assignment problem, we must ensure that the service requirement is met by the route that is chosen *subsequently* by the routing problem. An additional complication involves computing the cost for each possible assignment. This cost depends on the route between the first DC node and the last DC node, which is determined by the routing subproblem. However, the routing subproblem is not defined until we obtain the

volumes from the assignment problem. Even then, we can obtain only an estimate of the assignment cost since the shipping cost on a link does not separate by origin-destination pair.

One approach to deal with this dilemma is to iterate between the two problems. Using an estimate of the routing component of the assignment cost, we solve the assignment problem. With aggregate volumes from the solution to the assignment problem, we solve the routing problem to obtain an exact routing cost. Thus, we have ‘priced out’ one possible assignment. Next, we update our estimates of the assignment costs using the solution from the routing problem, and re-solve the assignment problem and the corresponding routing problem. We iterate back and forth between the two subproblems. We can view this iterative scheme as an heuristic search over the possible assignments. In our computational study, we have focused on testing the algorithms for each subproblem, and thus not tested the iterative scheme.

In the remainder of this paper, we will present mathematical formulations for both the assignment and routing subproblems, discuss optimization-based solution approaches for each subproblem as well as a heuristic method for the routing subproblem, and present some computational results on a large scale problem met in practice.

## 4 The Assignment Subproblem

Instead of determining the complete route for each origin-destination terminal pair, the assignment subproblem focuses on assigning only the first and last DC nodes for each origin-destination terminal pair.

The assignment subproblem can be formulated as follows:

**Problem AP:**

$$Z^* = \min_{z, v, w} \sum_{ij} v^i d_{ij} x_{ij} + \sum_h s'_h w_h + \sum_l c_l y_l$$

subject to

$$\sum_j x_{ij} = 1 \quad \forall \text{ O-D pairs } i \quad (1)$$

$$\sum_{ij} v^i x_{ij} \leq C_h + w_h \quad \forall \text{ DC nodes } h \quad (2)$$

$$\sum_{ij} v^i x_{ij} \leq K y_l \quad \forall \text{ links } l \quad (3)$$

$$0 \leq x_{ij} \leq 1 \text{ integer} \quad \forall ij \quad (4)$$

$$0 \leq w_h \leq w_h^{\max} \quad \forall \text{ DC nodes } h \quad (5)$$

$$y_l \geq 0, \text{ integer} \quad \forall \text{ links } l. \quad (6)$$

The decision variables for the model are

$$x_{ij} = \begin{cases} 1 & \text{if O-D pair } i \text{ is assigned to DC-pair } j, \\ 0 & \text{otherwise} \end{cases}$$

$y_l$  = number of trailers dispatched on link  $l$ , and

$w_h$  = excess capacity at DC node  $h$  (beyond the nominal capacity).

The cost and constraint parameters are

- $v^i$  = volume for O-D pair  $i$ ,
- $d_{ij}$  = assignment cost for assigning O-D pair  $i$  to DC pair  $j$   
 = (processing cost per unit at the two end DC nodes)  
 + (estimated DC-to-DC shipping and processing cost)
- $c_l$  = cost of dispatching one trailer on link  $l$ ,
- $s'_h$  = additional processing cost per unit for excess items at DC node  $h$ ,
- $K$  = capacity of a trailer,
- $C_h$  = processing capacity at node  $h$ , and
- $w_h^{\max}$  = maximum excess capacity at DC node  $h$ .

Constraints (1) and (4) of this problem (which we refer to as *AP*) force each O-D pair to be assigned to one DC node pair. The assignments  $ij$  need not include all possible assignments, but can be pre-selected to eliminate those that violate service-time restrictions. Constraint (2) ensures that sufficient excess capacity is allocated at the distribution centers to cover the volume processed. This constraint, together with the objective function, models (as a piecewise linear approximation) the nonlinear DC cost function shown in Figure 3. The lefthand side of the constraint is summed over all assignments  $ij$  for which node  $h$  is one of the nodes in the DC-node pair  $j$ .

Constraint (3) ensures that the volume on the link  $l$  can be carried by the number of trailers dispatched on that link. This constraint, together with the objective function coefficients of  $y_l$  models the link cost step-function shown in Figure 2. Note that in this model, we are assuming a homogeneous fleet of trailers with a fixed capacity  $K$ . Therefore, on each link the cost step-function has jumps of equal size which occur at every increment  $K$  of capacity.

The assignment subproblem focuses only on the assignment of a DC-node pair to each O-D terminal pair and the dispatch of trailers on each terminal-DC and DC-terminal link, but does not deal with the dispatch of trailers on the intermediate DC-to-DC links. The two sets of decisions must be compatible so that the demand between the origin and destinations can be carried with the number of trailers dispatched on the end links.

This assignment problem does not explicitly model the intermediate shipping and processing costs, but uses a linear approximation, the ‘assign-

ment' cost  $d_{ij}$ , of the intermediate transportation costs. For each DC node, there is a nominal per unit processing cost, assuming that the processing facilities at the DC nodes are operating at the efficient levels (see Figure 3). Similarly, there is a nominal per unit shipping cost for each link assuming all trucks sent along a link carry full loads. These costs are linear approximations of the true costs. The assignment cost  $d_{ij}$  includes the nominal processing costs at the first and last DC node and the transportation cost between these two DC's along the cheapest path according to the nominal costs. These costs do not reflect any additional cost due to congestion at the processing facilities. The loads on the DC-DC links are typically large, which means that a linear approximation to the transportation cost should be acceptable. On the other hand, since the terminal-DC link loads are typically small, it is important to model accurately the 'step-function' nature of the costs.

## 4.1 Solution Method

Our solution approach to this problem iterates between a trailer allocation problem and an (O-D pair)-to-(DC pair) assignment problem. We first fix the maximum number of trailers that can be dispatched on each terminal-distribution center link. Then we try, using a Lagrangian relaxation approach, to find the cheapest assignment that uses no more than the allowable number of trailers on each link. This approach has the benefit of providing price information from the assignment solution that suggests how to adjust the allocation of trailers for each link. We then adjust the trailer allocation on several of the terminal-DC links and again try to find a compatible assignment. After repeating these steps several times, we choose as the final solution the assignment with the minimum overall cost.

Once the number of trailers for a link has been fixed, we effectively have a capacity constrained assignment problem. Thus, once we fix  $y$ , say to  $\tilde{y}$ ,  $AP$  for this value  $\tilde{y}$  becomes the following problem:

**Problem  $AP(\tilde{y})$ :**

$$Z^*(\tilde{y}) = \sum_l c_l \tilde{y}_l + \min_{x,w} \sum_{ij} v^i d_{ij} x_{ij} + \sum_h s'_h w_h$$

subject to

$$\sum_j x_{ij} = 1 \quad \forall \text{ O-D pairs } i \quad (7)$$

$$\lambda : \quad \sum_{ij} v^i x_{ij} \leq C_h + w_h \quad \forall \text{ DC nodes } h \quad (8)$$

$$\alpha : \quad \sum_{ij} v^i x_{ij} \leq \min\{m_l, K \tilde{y}_l\} \quad \forall \text{ links } l \quad (9)$$

$$0 \leq x_{ij} \leq 1 \quad \text{integer} \quad \forall ij \quad (10)$$

$$0 \leq w_h \leq w_h^{\max} \quad \forall \text{ DC nodes } h \quad (11)$$

where

$m_l =$  maximum volume on link  $l$  based on demand requirements.

#### 4.1.1 Solution of $AP(\tilde{y})$ by Lagrangian Relaxation

We “solve”  $AP(\tilde{y})$  by Lagrangian relaxation: dualizing constraints (8) and (9) produces the following relaxed problem,  $\mathcal{L}(\alpha, \lambda; \tilde{y})$ , for a given set of values for the Lagrangian dual variables  $\alpha, \lambda$  and trailer assignments  $\tilde{y}$ :

$$L(\alpha, \lambda; \tilde{y}) = \sum_l c_l \tilde{y}_l - \sum_h \lambda_h C_h - \sum_l \alpha_l \min\{m_l, K \tilde{y}_l\} + V(\alpha, \lambda).$$

The final term in this expression represents the optimal value of the optimization problem



**Problem  $\mathcal{L}(\alpha, \lambda; \tilde{y})$ :**

$$V(\alpha, \lambda) = \min_{x, w} \left\{ \sum_{ij} v^i (d_{ij} + \alpha_{l1} + \alpha_{l2} + \lambda_{h1} + \lambda_{h2}) x_{ij} + \sum_h (s'_h - \lambda_h) w_h \right\}$$

subject to

$$\sum_j x_{ij} = 1 \quad \forall \text{ O-D pairs } i \quad (12)$$

$$0 \leq x_{ij} \leq 1 \text{ integer} \quad \forall ij \quad (13)$$

$$0 \leq w_h \leq w_h^{\max} \quad \forall \text{ DC nodes } h. \quad (14)$$

The indices  $h1$  and  $h2$  refer respectively to the first and last DC of the DC pair  $j$ . Similarly, the index  $l1$  refers to the link from the origin of the O-D pair  $i$  to the first DC of the DC pair  $j$  and  $l2$  refers to the link from the last DC of DC pair  $j$  to the destination of the O-D pair  $i$ .

The problem  $\mathcal{L}(\alpha, \lambda; \tilde{y})$  is easy to solve. For each  $i$ , we simply set  $x_{ij} = 1$  for the index  $j$  with the smallest cost coefficient of  $x_{ij}$ . For each  $h$ , we set  $w_h$  to its upper (respectively lower) bound according to whether the sign of its coefficient in the objective function is negative (respectively positive).

The elementary “weak duality” result from the theory of Lagrangian relaxation (see Shapiro [1979b]) shows that  $L(\alpha, \lambda; \tilde{y})$  is a lower bound on the objective value  $Z^*(\tilde{y})$  of the problem  $AP(\tilde{y})$ . We are interested in obtaining the tightest lower bound on  $Z^*(\tilde{y})$ , i.e., we would like to solve for

$$L^*(\tilde{y}) = \max_{\substack{\alpha \geq 0 \\ \lambda \geq 0}} L(\alpha, \lambda; \tilde{y}) \leq Z^*(\tilde{y}). \quad (15)$$

Our implementation uses a subgradient optimization approach (see Shapiro [1979b]) to solve (15). Since  $Z^*(\tilde{y})$  is unknown, an upper bound on it is used in computing the subgradient stepsize. Thus, convergence is not guaranteed. In our implementation, we perform subgradient iterations until

1. *either* the duality gap is smaller than 1 %,
2. *or* the algorithm has taken a pre-set number of subgradient steps.

### 4.1.2 Updating of $y$

When the Lagrangian dual problem  $L^*(\tilde{y})$  has been solved, the final values of the Lagrangian multipliers will indicate the degree to which the corresponding capacity constraints are binding. A Lagrangian multiplier  $\alpha_l$  of zero indicates that the constraint is not binding, and suggests that it is possible to decrease the allowable number of trailers on link  $l$ . On the other hand, a large Lagrangian multiplier indicates that the corresponding capacity constraint is tight and suggests that we should increase the allowable number of trailers.

We have implemented a heuristic scheme for updating the value of  $y$  from iteration to iteration. The change in  $y$  is based on the value of  $L^*(\tilde{y})$  (or our best estimate of its value). Notice that, in terms of the solution  $(\alpha^*, \lambda^*)$  of the dual problem (15),  $L^*(\tilde{y})$  can be expressed as

$$L^*(\tilde{y}) = \dots + \sum_{m_l > K\tilde{y}_l} (c_l - \alpha_l^* K)\tilde{y}_l + \sum_{m_l \leq K\tilde{y}_l} c_l\tilde{y}_l + \dots \quad (16)$$

We try to *increase*  $\tilde{y}_l$  if its coefficient in  $L^*(\tilde{y})$  is negative and *decrease*  $\tilde{y}_l$  if its coefficient is positive. In our implementation, we first tried changing all the  $\tilde{y}_l$ 's simultaneously, but the new set of  $\tilde{y}$  values were often infeasible for the original problem (in that they did not provide sufficient transport capacity). Thus, to avoid infeasibility, we change the value of  $\tilde{y}$  for only a few of the links and increment or decrement each  $\tilde{y}_l$  by only one unit. At each iteration, we always try to increment a pre-set number (e.g., 40) of  $\tilde{y}_l$ 's. The implementation examines the  $\tilde{y}_l$ 's eligible to be incremented in increasing order of their coefficients, and increments each unless it is currently at its upper bound. If no  $\tilde{y}_l$  can be increased, we try to decrease the value for a (possibly different) pre-set number (e.g. 5) of  $\tilde{y}_l$ 's. In this case, the implementation examines the  $\tilde{y}_l$ 's in decreasing order of their coefficients, and decrements each unless it is already at its lower bound.

The rationale for changing the value of  $\tilde{y}_l$  based on the sign of its coefficient in  $L^*(\tilde{y})$  can be described as follows. We know that  $Z^*(\tilde{y})$  is a piecewise linear function of  $\tilde{y}$ . For a given value of  $\tilde{y}$ ,  $L^*(\tilde{y})$  is a lower approximation of  $Z^*(\tilde{y})$ . Thus, we are using  $L^*(\tilde{y})$  as an approximation of

the function  $Z^*(\tilde{y})$ . Hence, our updating procedure is a local improvement scheme, except that it uses the subgradient of  $L^*(\tilde{y})$  at  $\tilde{y}$  as a surrogate for the unknown subgradient of  $Z^*(\tilde{y})$  at  $\tilde{y}$ .

### 4.1.3 Bounds on $Z^*$

For fixed  $\alpha$  and  $\lambda$ , the solution of the Lagrangian problem does not depend on the value of  $\tilde{y}$ . Hence for any given  $\alpha \geq 0$  and  $\lambda \geq 0$ , if  $\lceil x \rceil$  denotes the smallest integer greater than or equal to  $x$ , we can define

$$LL(\alpha, \lambda) = \sum_{c_l < 0} c_l \lceil \frac{m_l}{K} \rceil - \sum_h \lambda_h C_h - \sum_{\alpha_l > 0} \alpha_l \lceil \frac{m_l}{K} \rceil + V(\alpha, \lambda)$$

by setting  $\tilde{y}$  to its lower (respectively upper) bound whenever its coefficient in  $L(\alpha, \lambda; \tilde{y})$  is positive (respectively negative). Now for any given  $\alpha \geq 0$ ,  $\lambda \geq 0$ ,

$$LL(\alpha, \lambda) \leq L(\alpha, \lambda; \tilde{y}) \leq L^*(\tilde{y}) \leq Z^*(\tilde{y}) \quad \text{for all } \tilde{y}.$$

Thus, in particular, if  $L(\alpha^*, \lambda^*; \tilde{y})$  is our approximation of  $L^*(\tilde{y})$ , then  $LL(\alpha^*, \lambda^*)$  gives a lower bound on the value of  $Z^*$ .

An upper bound on  $Z^*$  is provided by the current best solution to  $AP$ . This best assignment can be found as we solve  $L^*(\tilde{y})$ . While any assignment generated by the solution of  $\mathcal{L}(\alpha, \lambda; \tilde{y})$  might not satisfy the capacity constraint imposed by a fixed allocation of trailers  $\tilde{y}$ , it is nonetheless a valid assignment for the problem  $AP$  for some value of  $y$ . In our implementation, we evaluate all the possible assignments considered irrespective of the current trailer allocation  $\tilde{y}$ , and retain the one with the minimum overall cost as the final solution.

Figure 4 shows a flow chart of the complete algorithm.

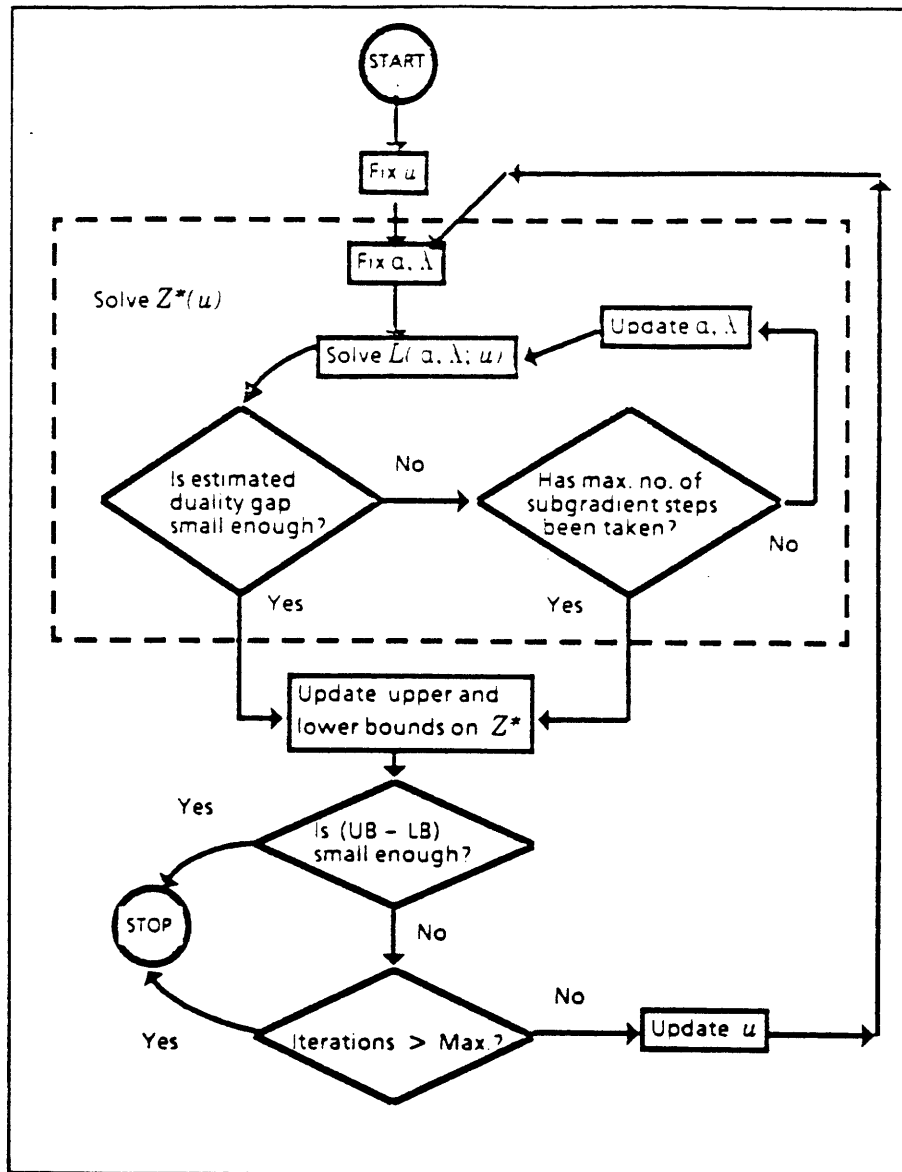


Figure 4: Flow Chart for the Assignment Algorithm

## 5 The Routing Problem

In this section, we will formulate the DC-DC routing problem and present a Lagrangian relaxation approach for solving it. We consider the following version of the routing problem:

**Problem (R):**

$$\min \sum_{ij} c_{ij} y_{ij} + \sum_{\text{DC node } i} s_i \sum_{\text{DC node } j} \sum_{ab} v^{ab} z_{ij}^{ab}$$

subject to

$$\sum_i z_{ij}^{ab} - \sum_k z_{jk}^{ab} = \begin{cases} 1 & \text{if } j = b \\ -1 & \text{if } j = a \\ 0 & \text{otherwise} \end{cases} \quad \forall j, \forall \text{O-D pairs } ab \quad (17)$$

$$\sum_{ab} v^{ab} z_{ij}^{ab} \leq K y_{ij} \quad \forall \text{links } ij \quad (18)$$

$$z_{ij}^{ab} \in \{0, 1\}$$

$$y_{ij} \geq 0 \text{ integer.}$$

The decision variables in this model are

$y_{ij}$  = number of trailers dispatched from DC  $i$  to DC  $j$ , and

$z_{ij}^{ab}$  =  $\begin{cases} 1 & \text{if demand for O-D pair (i.e. DC pair) } ab \text{ flows from DC } i \text{ to DC } j \\ 0 & \text{otherwise} \end{cases}$

The parameters are

$v^{ab}$  = volume (demand) between DC pair  $ab$ ,

$c_{ij}$  = cost of dispatching a trailer from DC  $i$  to DC  $j$ ,

$s_i$  = cost of processing one unit of demand at DC node  $i$ , and

$K$  = capacity of a trailer.

The volume  $v^{ab}$  between DC pair  $ab$  depends upon the assignment of O-D terminal pairs  $i$  to DC pairs  $ab$  as specified in the assignment problem considered in the previous section (i.e.,  $v^{ab}$  equals the sum of volumes  $v^i$  for all terminal pairs  $i$  assigned to the DC pair  $ab$ ).

This model ignores the distribution center capacity and timing constraints. In this formulation, we have also assumed that the processing cost per item is constant for each distribution center. The problem  $R$  is essentially a multi-commodity flow problem that distinguishes the commodities by their origin-destination pair. The complication of the problem is that the link cost is not proportional to the flow on the link, and hence cannot be separated by commodities.

Note that when applied to the overall delivery planning problem, this route planning problem would be large: for our application with 2000 links and 8000 terminal pairs, it would contain 16 million binary variables  $z$ . The decomposition approach we have adopted has reduced this number considerably; yet with 40 DC's, it will still contain  $40^4 = 2.56$  million of these variables.

For the problem  $R$ , we have a choice of dualizing either constraints (17) or (18). By dualizing the constraints (18), we obtain a lower bound on the optimum value which equals that obtained from the LP-relaxation of  $R$ . (See Fisher [1981] or Geoffrion [1977].) The LP-duality gap for vehicle fleet planning problems may often be large, and we would like to obtain a Lagrangian relaxation that gives a tighter lower bound. Therefore, we choose to dualize constraints (17) instead. The corresponding Lagrangian problem then becomes:

$$L(\lambda) = \min \sum_{ij} c_{ij} y_{ij} - \sum_{ij} \sum_{ab} (\lambda_j^{ab} - \lambda_i^{ab} - s_i) v^{ab} z_{ij}^{ab} + \sum_b \lambda_b^{ab}$$

subject to

$$\begin{aligned} \sum_{ab} v^{ab} z_{ij}^{ab} &\leq K y_{ij} && \forall \text{ links } ij \\ z_{ij}^{ab} &\in \{0, 1\}, && y_{ij} \geq 0, \text{ integer.} \end{aligned} \tag{18}$$

Again we try to obtain the best possible Lagrangian lower bound by solving the Lagrangian dual problem:

$$L^* = \max_{\lambda} L(\lambda).$$

Our hope is that dualizing (17) instead of (18) will close some of the LP-relaxation duality gap; for several applications, our computational results seem to bear out this prospect.

Notice also that the Lagrangian problem  $L(\lambda)$  separates into a knapsack-like subproblem for each link. Thus  $L(\lambda) = \sum_{ij} L_{ij}(\lambda) + \sum_b \lambda_b^{ab}$  where

$$L_{ij}(\lambda) = \min c_{ij} y_{ij} - \sum_{ab} (\lambda_j^{ab} - \lambda_i^{ab} - s_i) v^{ab} z_{ij}^{ab}$$

subject to

$$\sum_{ab} v^{ab} z_{ij}^{ab} \leq K y_{ij} \tag{19}$$

$$z_{ij}^{ab} \in \{0, 1\}, \quad y_{ij} \geq 0, \text{ integer.}$$

We can take advantage of this observation and solve  $L(\lambda)$  by solving a sequence of subproblems, one for each link in the network. Changing the  $\lambda_j^{ab}$ 's individually permits us to exploit this advantage computationally; a change in  $\lambda_j^{ab}$  induces a change in  $L_{ij}(\lambda)$  for only the links adjacent to node  $j$ .

## 5.1 Solution of the Dual Problem by Multiplier Adjustment

We propose a multiplier adjustment procedure that achieves dual ascent (i.e., never diminishes the value of  $L(\lambda)$ ) and that provides a mechanism for constructing feasible solutions to the original problem  $R$ .

Suppose  $(\tilde{y}, \tilde{z})$  solves the Lagrangian problem  $L(\tilde{\lambda})$  for some  $\tilde{\lambda}$ . If the  $\tilde{z}_{ij}^{ab}$ 's define exactly *one* path from  $a$  to  $b$  for each  $ab$ , then  $(\tilde{y}, \tilde{z})$  is an optimal solution to our original problem  $R$ . Otherwise, we want to construct a feasible solution to  $R$  using the values of  $\tilde{z}_{ij}^{ab}$ .

We next describe how to construct a path, using solutions to the Lagrangian relaxation, for a given origin-destination pair  $ab$ . For the given  $ab$ , let

$$E^{ab} = \{ij \mid \tilde{z}_{ij}^{ab} = 1\}$$

be the set of “chosen” links for the origin-destination pair (obtained from solving for the  $L_{ij}(\lambda)$ ). If there is a path from  $a$  to  $b$  using only the “chosen” links of the network, we let this path be the assigned route from  $a$  to  $b$ . If such “chosen” paths can be found for all  $ab$ , they constitute a feasible solution to the problem  $R$ .

Suppose there is no “chosen” path from  $a$  to  $b$ . Let  $\mathbf{R}$  be the set of nodes *reachable* from  $a$  along only links in  $E$ . The network might look like the diagram in Figure 5.

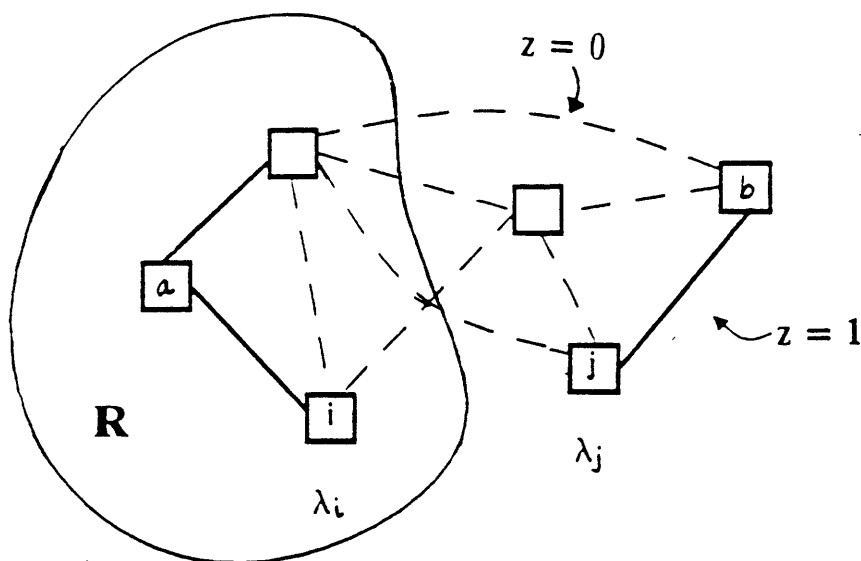


Figure 5: Network with no *chosen* path from  $a$  to  $b$ .

In order for there to be a “chosen” path from  $a$  to  $b$ , we want the value of  $z_{ij}^{ab}$  on some link in the cutset to be set to one, which would be so if the coefficient of  $z_{ij}^{ab}$  in the objective function of  $L(\lambda)$  is a large enough negative number. Hence, we want to increase  $\tilde{\lambda}_j^{ab}$  for some node  $j \notin \mathbf{R}$ . Define

$\Delta_{ij}^{ab}$  = minimum increase in  $\tilde{\lambda}_j^{ab}$  so that  $z_{ij}^{ab} = 1$  in the Lagrangian problem  $L(\lambda)$ .

Let

$$\Delta^{ab} = \min \{ \Delta_{ij}^{ab} \mid i \in \mathbf{R}, j \notin \mathbf{R} \}$$



and let

$$i'j' = \underset{ij}{\operatorname{argmin}} \Delta_{ij}^{ab}.$$

At this point, we increase the value of  $\tilde{\lambda}_j^{ab}$  by  $\Delta^{ab}$  for every node  $j \notin \mathbf{R}$ . By the definition of  $\Delta^{ab}$ , this change affects the Lagrangian subproblem only for the link  $i'j'$ . Hence, we need to re-solve only the Lagrangian subproblem for one link ( $i'j'$ ) to obtain the new solution  $L(\lambda)$  for the new values of  $\lambda$ .

Because of the way  $\Delta^{ab}$  is chosen, it is not difficult to see that the new value of  $L_{i'j'}(\lambda)$  must be no less than its previous value  $L_{i'j'}(\tilde{\lambda})$ . Since  $\lambda_b^{ab}$  increases by  $\Delta^{ab}$ , the value of  $L(\lambda)$  increases by at least  $\Delta^{ab}$ .

We can repeat this multiplier adjustment procedure as we probe for a path from  $a$  to  $b$ . When the method has constructed a path of “chosen” links, it has simultaneously constructed a feasible route from  $a$  to  $b$  and also obtained a tighter lower bound for  $L^*$ .

Our solution procedure constructs a path for each  $ab$  in turn in a pre-defined order. As we probe for a path from  $a$  to  $b$ , we update the values of  $\tilde{\lambda}_j^{ab}$ 's and the solutions corresponding to the  $L_{ij}(\tilde{\lambda})$ 's, changing the complete flow pattern at each updating. Because the values of  $z_{ij}^{ab}$  may change when the subproblems corresponding to the  $L_{ij}(\lambda)$  are re-solved, the set of paths found so far may not be compatible with the current set of values of  $z_{ij}^{ab}$ . Nonetheless, the values of  $z_{ij}^{ab}$  represent a feasible solution to the routing problem. Our implementation examines the list of the origin-destination pairs  $ab$  twice. Since the algorithm modifies the complete flow pattern, the path for a given  $ab$  may change in the second pass through the list of O-D pairs. The number of passes through the list of O-D pairs is a parameter to be set heuristically.

### 5.1.1 Solution of the Lagrangian Problem $L(\lambda)$

As indicated in the earlier part of this section, the Lagrangian problem (to find  $L(\lambda)$ ) separates into a knapsack-like problem (to find  $L_{ij}(\lambda)$ ) for each

link  $ij$  of the network. Although an exact solution by dynamic programming is possible, to avoid costly computation, we ‘solve’ this problem by using a greedy heuristic. We compute

$$\bar{y} = \left\lfloor \frac{\sum v^{ab}}{S_{ij} K} \right\rfloor \quad \text{where} \quad S_{ij} = \left\{ ab \mid \frac{\lambda_j^{ab} - \lambda_i^{ab} - s_i}{v^{ab}} \geq \frac{c_{ij}}{K} \right\}.$$

We then solve for  $L_{ij}(\lambda)$  by using a greedy heuristic for the knapsack problem with the righthand side of the constraint (19) fixed at either  $K\bar{y}$  or  $K(\bar{y} - 1)$ . The better of the two solutions is chosen.

## 5.2 A Lagrangian Relaxation Method for the Routing Problem

Let us now summarize the overall scheme to solve the routing problem. We start with an arbitrary assignment of  $\lambda_j^{ab}$  and solve the Lagrangian relaxation problem  $L(\lambda)$ . The following choice is a useful initial assignment of the  $\lambda_j^{ab}$ 's:

For each  $ab$ , set  $\lambda_a^{ab} = 0$  and let  $\lambda_j^{ab} = \lambda_i^{ab} + s_i v^{ab} + \frac{v^{ab}}{K} c_{ij}$  for all other nodes  $j$ , where link  $ij$  is on the cheapest path from  $a$  to  $j$  with respect to the linearized cost  $s_i v^{ab} + \frac{v^{ab}}{K} c_{ij}$ .

Such an assignment of the  $\lambda_j^{ab}$ 's has the property that for any node  $j$  in the network,  $\lambda_j^{ab}$  is the cheapest per unit linearized cost of shipping goods from node  $a$  to node  $j$ . Thus if link  $ij$  is on a cheapest path (with respect to the linearized cost) from node  $a$  to node  $b$ ,  $\frac{(\lambda_j^{ab} - \lambda_i^{ab} - s_i v^{ab})}{v^{ab}} - \frac{c_{ij}}{K}$  is zero and consequently  $z_{ij}^{ab}$  is likely to be set to 1 in the solution for the knapsack subproblem  $L_{ij}(\lambda)$  for link  $ij$ .

The solution of the Lagrangian problem gives an lower bound on the optimum value of  $R$ . With the Lagrangian solution we can also construct a feasible solution to  $R$  by probing for a route for each origin-destination pair  $ab$  by the method described in Section 5.1. The value of the best feasible

solution found, call it  $\mathbf{V}_F$ , provides an upper bound on the optimum value of  $R$ . We can continue the multiplier adjustment procedure, simultaneously generating feasible solutions and increasing the value of  $L(\lambda)$ , until

$$\frac{\mathbf{V}_F - L(\lambda)}{L(\lambda)} \leq \epsilon \quad (20)$$

where  $\epsilon$  is some pre-determined bound.

Since we are using a heuristic solution procedure to solve the subproblem  $L_{ij}(\lambda)$ , we are overestimating the value of  $L(\lambda)$ , and so the expression on the left hand side of (20) slightly underestimates the gap between the best feasible solution found and  $L(\lambda)$ . Moreover, because we solve  $L(\lambda)$  approximately, every change in  $\lambda$  does not guarantee dual ascent.

## 6 Marginal cost Heuristic for the Route Planning Problem

In addition to the optimization-based method described in the previous sections, we also investigated heuristic approaches to the route planning problem.

One heuristic, called the Marginal Cost Heuristic, tries to improve one route at a time, according to the ‘marginal’ cost of using a link. To represent both the DC processing costs and link transportation costs as edge costs on a network, we construct an extended network with each DC node split into two nodes with an adjoining edge. The cost of this edge represent the DC processing cost. (See Figure 6.)

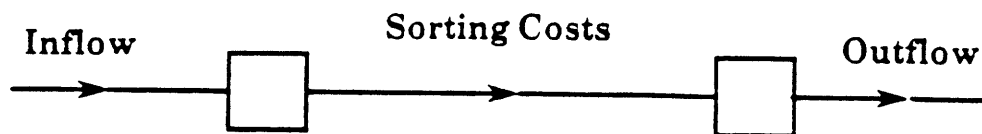


Figure 6: ‘Splitting’ DC node to form extended network

The Marginal Heuristic proceeds by sequentially trying to improve the route for one origin-destination terminal pair at a time. The current route plan is taken to be the initial solution. We then examine each O-D terminal pair in turn. We first remove the volume from the current path for this O-D pair, say,  $ab$ . Then for each edge, we compute the *marginal* cost of sending the volume for this O-D pair on each edge, as follows.

If the edge is one adjoining two ‘split’ DC nodes, then

$$\text{Cost on edge } d = \text{PCost}(d, f + v) - \text{PCost}(d, f)$$

where

$v$  = volume for O-D pair  $ab$ ,  
 $f$  = flow through DC node  $d$  due to the current path assignments for all paths except  $ab$ ,

$\text{PCost}(d, g)$  = total processing cost for DC node  $d$  when the flow through the node is  $g$ .  

$$= \begin{cases} p_d g & \text{for } C_d^{\min} \leq g \leq C_d^{\max} \\ p_d(1 + 0.04(g - C_d^{\max}))g & \text{for } g \geq C_d^{\max} \\ p_d(1 + 0.01(C_d^{\min} - g))g & \text{for } g \leq C_d^{\min} \end{cases}$$

$p_d$  = nominal processing cost per item at DC node  $d$ ,  
 $C_d^{\min}$  = minimum capacity at DC node  $d$ , and  
 $C_d^{\max}$  = maximum capacity at DC node  $d$ .

The function  $\text{PCost}(d, g)$  is an approximation that reflects the non-linearity of the processing cost at the DC node as indicated by Figure 3.

If the edge represents an actual link in the network, then

$$\text{Cost on link } l = \text{Cost}(l, f) - \text{Cost}(l, f + v) + \theta \text{Time}(l)$$

where

$f$  = flow on link  $l$  due to the current path assignments for all paths except  $ab$ ,  
 $\theta$  = weighting constant (pre-set parameter),  
 $\text{Time}(l)$  = Time interval between the activities at the two end-nodes of link  $l$ , and  
 $\text{Cost}(l, g)$  = total transportation cost on link  $l$  when the flow on the link is  $g$ .

After computing all the costs of the edges, the heuristic applies Dijkstra's [1962] algorithm and finds a 'shortest' path between the O-D pair  $ab$ . If this new path improves upon the current path (with respect to actual cost), the new path is made the current path and the volume  $v$  for the O-D pair  $ab$  is added to all edges on the new path. We then repeat this entire procedure for each O-D pair in turn, making changes in the current paths whenever the method finds an improvement. If, after cycling through all O-D pairs, the improvement obtained is too small, the algorithm stops.

Otherwise, it again starts by examining the first O-D pair and repeats the cycle. It also stops after a maximum number of cycles. Figure 7 gives a flow chart for this heuristic.

Notice that the edge cost used in Dijkstra's algorithm is a weighted combination of the transportation cost and the transportation time. This approach attempts to incorporate the service requirements into our model. (A similar approach would permit us to use this costing procedure in the routing model  $R$  to account for service requirements.) We have found that for our application, with a judicious choice of the weighting parameter ( $\theta = 5$ ), the heuristic generates routes that are cost effective and contain very few service violations.

Singhal [1984] provides more details on the marginal cost heuristic.

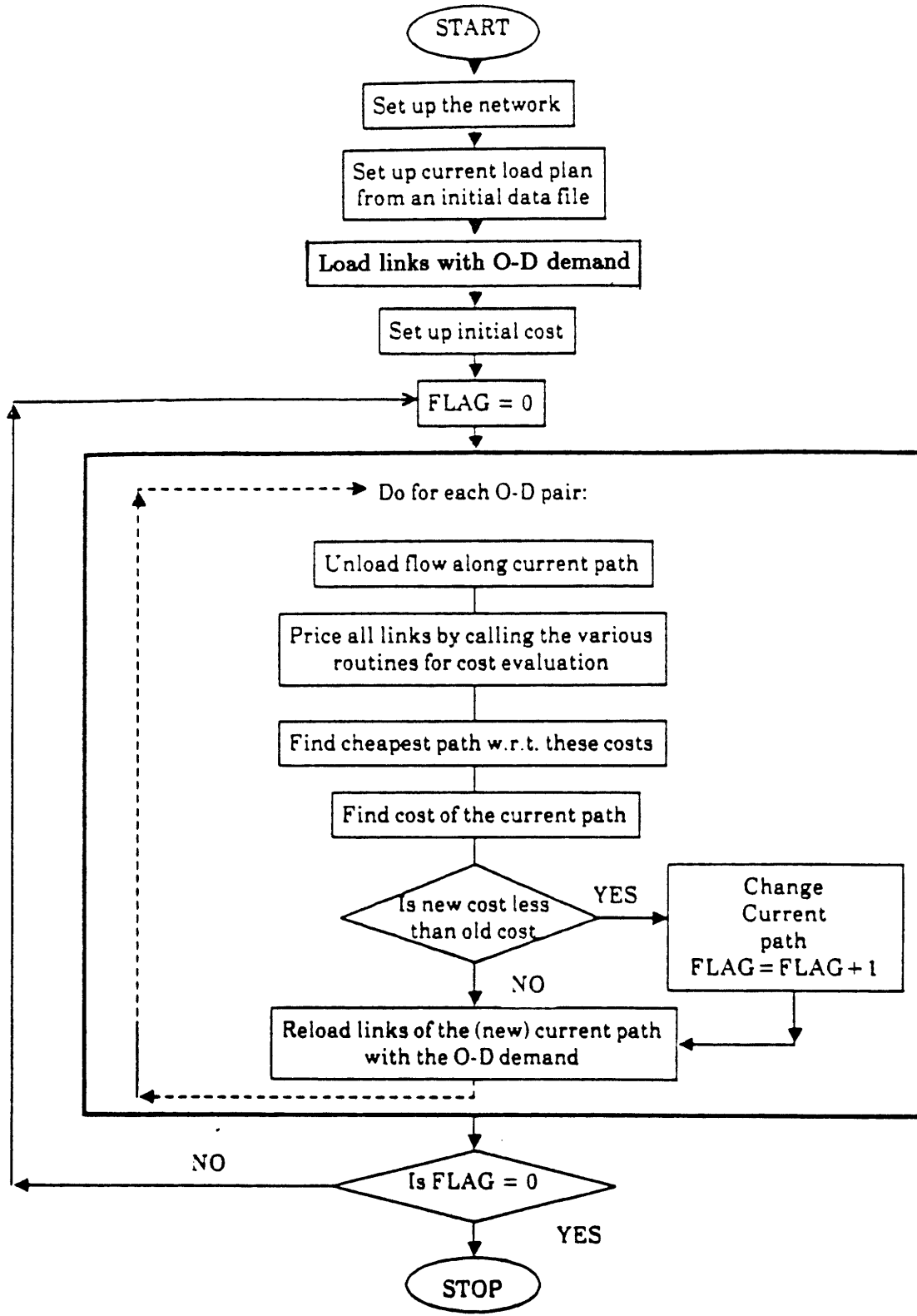


Figure 7: Flow chart for the Marginal Cost Heuristic

## 7 Computational Results

### 7.1 The Assignment Algorithm

We tested the algorithm for the assignment problem as described in Section 4 on a delivery application (with real data) with 211 terminal nodes, 36 distribution centers and 7976 origin-destination pairs for a large-scale transportation company. We pre-processed the data to compile a list of “feasible” DC-node pair assignments for each O-D pair. Assignments that clearly violate the timing requirements for service were not included in the list. For the problem we studied, the average number of feasible assignments was 3. Table 1 shows the computational results of the assignment algorithm for two runs on the same physical network using slightly different cost structures. The data have been rescaled to facilitate comparisons.

Table 1 contains the following entries:

1. The initial solution  $Z_I$  is a normalized ‘assignment cost’ of the current operating plan,
2. The final solution with ‘assignment cost’  $Z_F$  is the best solution generated by the assignment algorithm,
3.  $Z_L$  is the value of the initial solution when costed with respect to the LP-relaxation of the problem  $AP$ ,
4. The lower bound on  $Z^*$  is the best bound from all the Lagrangian problems as computed according to Section 4.1,
5. The average Lagrangian duality gap is the average discrepancy between the best solution found for a fixed  $\tilde{y}$  and the highest Lagrangian value (The entry is the average for all Lagrangian problems for  $\tilde{y}$  where  $Z^*(\tilde{y})$  is finite.),
6. The integrality gap estimates the discrepancy between the optimum value of  $AP$  and its LP-relaxation; since  $Z_L \geq Z_{LP}$ ,  $Z_{LP}$  being the



Dataset		ASG1	ASG2
Initial Solution	$Z_I$	1000.00	1000.00
Best Solution	$Z_F$	989.86	989.48
Initial LP Value	$Z_L$	815.33	837.55
Lower Bound on $Z^*$	$Z_{LB}$	583.71	572.46
Average Lagrangian Duality Gap	$\frac{Z(\bar{y}) - L^*(\bar{y})}{L^*(\bar{y})}$ (est)	3.42%	1.80%
Integrality Gap	$\frac{Z_F - Z_L}{Z_L}$	21.41%	19.40%
Savings	$\frac{Z_I - Z_F}{Z_I}$	1.02%	1.06%
Total no. of iterations		10	10
Iteration when best solution found		2	2
CPU Time (Prime 850)	(minutes)	39.70	47.53

Table 1: Computational Results – Assignment Algorithm

optimum value of the LP-relaxation, this entry may underestimate the integrality gap,

7. The savings is the percentage reduction in cost of the best solution found as compared to the initial solution,
8. The number of iterations is the number of times the value of  $y$  was updated during the run,
9. The next entry indicates when the best solution,  $Z_F$ , was found, and
10. The last entry gives the CPU time on a Prime 850 computer.

Notice that the duality gap for the Lagrangian problem (for fixed  $y$ ) is

quite small. On the other hand the difference between the optimum value of  $AP$  and its LP-relaxation, i.e., the integrality gap, is quite large. A naive approach to the assignment problem may be to solve the LP-relaxation and then round up the values of the integer variables. Our computational experiments indicate that this heuristic method yields very poor results for the delivery route planning problem; the actual costs of the route plans it generated were often more than 10% higher than the initial solution.

The large integrality gap is common for problems containing a step function or fixed-charge costs. Any relaxation algorithm for these problems must retain some of the integral structure in order to obtain tight bounds. Our algorithm attempts to do so by trying to fix  $y$  at judiciously chosen (and hopefully near-optimal) values. The results in Table 1 also suggest that a tighter estimate of the lower bound for  $Z^*$  is needed.

The initial solution represents the current operating plan of the company that supplied the data. This plan is perceived to be fine-tuned and large improvements were not expected. As Table 1 shows, the assignment algorithm seems to make only very modest improvements.

However, the ‘assignment cost’ for this problem uses a linear approximation for the intermediate shipping and DC processing cost, which does not accurately reflect the actual operating costs. Combined with the routing heuristic, the improvements to the overall routing plan is actually much better than the results indicated by the assignment algorithm. Table 2 is a breakdown of the costs for the dataset ASG2. The numbers in parenthesis indicate the percentage reduction in costs. For this dataset, the assignment algorithm reduces the DC processing cost at the expense of increasing the transportation cost on the terminal-DC links. The routing heuristic further reduces the intermediate transportation cost, giving an overall (actual) cost savings of almost 5%, instead of the 1% indicated by the assignment algorithm alone. Since the initial solution used in our computation is perceived to be close to the optimum, neither we nor our industrial collaborators expected savings of more than 5%.

		Actual Cost	Assignment Cost
Initial Solution:	Terminal-DC links	400.37	400.37
	Intermediate links and processing	703.06	599.63
	Total	1103.43	1000.00
Best Solution:	Terminal-DC links	411.15 (-2.69%)	411.15 (-2.69%)
	Intermediate links and processing	638.97 (9.12%)	578.33 (3.55%)
	Total	1050.12 (4.83%)	989.48 (1.06%)

Table 2: Cost Comparison for dataset ASG2.

## 7.2 The Routing Algorithm

We tested the algorithm for the problem  $R$  as outlined in Section 5 on six ‘real’ data sets. The data sets represented problems of different sizes, as shown in Table 3.

Dataset	NET1A	NET1B	NET1C	NET2A	NET2B	NET3
No. of nodes	12	40	43	36	36	36
No. of links	89	317	381	848	848	848
No. of O-D pairs	20	130	515	111	347	310

Table 3: Size of Test Problems

The first three data sets were extracted from the same real database. The underlying network and costs are the same, but the number of O-D pairs and the volume to be shipped between them are different. The next two data sets contain different demand patterns, but identical network structure and costs. The last data set NET3 represents a different network with a different cost structure.

Our algorithm for the routing problem  $R$  has been implemented in FOR-

TRAN77 on a Prime 850 mini-computer. The algorithm combines both subgradient optimization and the multiplier adjustment procedure as described in Section 5.

The experience with the test problems suggests that subgradient optimization consumes large amounts of computation time (over 60%) but does not improve the Lagrangian lower bound by much. Nonetheless, incorporating the subgradient steps in the algorithm is useful as a method of generating a new starting point for the multiplier adjustment procedure.

Table 4 summarizes the results of the algorithm on the six test problems. The costs have been re-scaled for ease of comparison.

Dataset		NET1A	NET1B	NET1C	NET2A	NET2B	NET3
Initial Cost	$V_I$	1000.00	1000.00	1000.00	1000.00	1000.00	1000.00
Final Cost	$V_F$	678.69	925.39	976.23	933.10	669.01	999.04
LP value	$V_{LP}$	428.75	685.58	797.23	699.50	476.33	472.22
Lower Bound	$\tilde{L}^*$	630.34	842.01	797.23	701.32	481.98	703.08
Duality Gap	$\frac{V_F - \tilde{L}^*}{\tilde{L}^*}$	7.67%	9.90%	22.45%	33.05%	38.80%	42.09%
Savings	$\frac{V_I - V_F}{V_I}$	38.84%	11.60%	1.58%	7.58%	32.91%	0.96%
Gap Reduction	$\frac{\tilde{L}^* - V_{LP}}{V_F - V_{LP}}$	80.65%	65.23%	0.00%	0.78%	2.93%	43.82%
CPU Time	(min)	2.08	97.45	117.81	50.25	110.34	90.92

Table 4: Computational Results-Routing Algorithm

Table 4 contains the following entries:

1. The initial solution is obtained by assigning to each O-D pair the route that is obtained from solving the LP-relaxation of  $R$  (i.e., the cheapest route according to the linearized cost). The cost is computed

exactly. The program also allows the option of starting with the current operation plan.

2. The best solution value is the cost of the best feasible solution generated in the course of the algorithm.
3. The next entry is the optimum value of the LP-relaxation of  $R$ .
4. The lower bound is the highest values of  $L(\lambda)$  generated in the course of the algorithm.
5. The duality gap is an estimate of the remaining duality gap at the termination of the algorithm.
6. The savings is the percentage reduction in cost of the best feasible solution found as compared to the initial solution.
7. The reduction in gap measures the percentage of the duality gap (as estimated by the difference between the best solution value found and the LP optimum value) that is closed by the Lagrangian lower bound.
8. The last row gives the computation time on the Prime 850.

The results are satisfactory in that the algorithm generates routes whose total cost is lower than routes generated according to a 'naive' linear cost approximation. For the smaller problems, the algorithm also closes a substantial portion of the duality gap, thus providing a much tighter bound on the optimum value of the problem  $R$ . This result seems to justify the decision to dualize constraints (17) instead of (18). However, the algorithm does not seem to make much headway in closing the gap for problems of larger size. The problem NET1C is particularly intransigent and the algorithm achieves no improvement at all over the linear programming relaxation bound. One explanation for this poor performance may be that the difference between an exact link cost and the linear approximation is small when the volume of flow on the links is large and the algorithm cannot make much improvement. However, if this is the case, then the duality gap ( $\mathbf{V}_F - \mathbf{V}_{LP}$ ) should also be small, but the results in Table 4 indicate

otherwise. More experimentation with the algorithm over a wider range of problems is necessary to understand its behavior.

In the computational tests, the first five data sets have the DC processing costs  $s_i$  fixed at zero, while the dataset NET3 contains nonzero values of  $s_i$ . The result on NET3 appears to be different that those of the other problems of comparable size. This result seems to suggest that the DC processing cost may have an important effect on the performance of the algorithm.

### 7.3 Marginal Cost Heuristic

The marginal cost heuristic was tested on a network with 36 DC nodes, 212 terminal nodes and 7976 origin-destination pairs. The current route plan was used as the initial solution. Table 5 gives the results from different runs of this heuristic using different values of the parameter  $\theta$  for weighting travel time.

	$\theta = 0$	$\theta = 5$	$\theta = 8$
Initial Cost	1000.0	1000.0	1000.0
Cost after Marginal Cost Heuristic	935.3	945.7	955.4
% savings	6.5	5.5	4.5
No. of service violations	462	9	6
Cost after post-processing	1001.4	952.0	963.5
% savings	-0.1	4.8	3.6
CPU-time (mins) (Prime 850)	58	187	249

Table 5: Computational Results for Marginal Cost Heuristic

When the timing considerations are ignored (i.e., when  $\theta = 0$ ), the marginal cost heuristic produced a 6.5% savings, but 462 origin-destination terminal pairs violated service time requirements. As  $\theta$  increases, the number of routes that violate service requirements decreases drastically. Our

programs also contains a post-processing stage that re-routes the volume for those origin-destination pairs that do not meet the required service timing levels. This is done by re-applying the heuristic (with  $\text{Time}(l)$  as link cost) for those origin-destination pairs whose assigned routes exceed the service time requirement. The CPU time given in Table 5 is the combined total for both stages.

## 7.4 Overall System

We have developed a comprehensive decision support system for the company whose problem motivated this research. This system includes the optimization algorithms and heuristics discussed in this paper as well as costing routines and an interactive network modification program (for adding and deleting DC's and links and changing demand assumptions and other parameters). In the solution approach used in the real delivery planning problem, we use the optimization approach we have described on the assignment problem and the marginal cost heuristic for the routing problem (as described in Section 6) instead of the routing algorithm described in Section 5. In this application setting, the intermediate routing cost comprises only 10% of the total cost and the routing heuristic works as well as the more time-consuming optimization.

Based on results from our computational testing, this modified optimization approach indicates a 4 to 5 percent cost improvement over the delivery route plan currently in use. Not all the indicated savings may be realizable, though, because operating rules may prevent the implementation of the routing plan generated by the optimization approach. Nonetheless, as we mentioned before, savings of this magnitude are substantial, representing profits of hundreds of thousands of dollars per year. Moreover, the company with which we have been working is noted for its efficiency; consequently, these cost savings indicate the potential of optimization methods in improving upon the schemes devised by even the most finely tuned planning systems that do not fully exploit modern mathematical programming methodology.

More importantly, the use of the decision support system facilitates delivery route planning by dramatically reducing the time required for a planning exercise. For the application that motivated this study, delivery route planning is done 18 months in advance (based on forecasted demand) and may take several weeks to complete. The algorithms presented in this paper can quickly generate a delivery routing plan for a given network configuration and demand pattern. This quick turnaround time enables the planning team to evaluate many more scenarios (with varying demand patterns and network configurations) and permits them to conduct a richer set of 'what-if' analysis.

A major disadvantage of heuristic approaches, as compared to optimization-based methods, is that they often do not provide an estimate of how far the final solution is from optimality. The optimization algorithms for both the assignment and routing problems presented in this paper provide upper and lower bounds on the optimum solution value. However, the computational results indicate that these bounds are not very tight. It would be useful to further analyze the models and algorithms and construct tighter bounds on the optimum solution value.

## 8 Conclusion

In this paper, we developed an optimization-based approach for a point-to-point route planning problem. The test data used represent only a part (but an important part) of the operations of the real-world problem that motivated this study, yet the resulting mixed-integer programming model is already prohibitively large. This fact, and the difficulty of solving the optimization models, highlight the discrepancy between mathematical programming theory and the practical needs of industry. The approach discussed in this paper exploits the structure of the problem to decompose it into two smaller subproblems, to which we can then apply optimization techniques independently. The computational results are encouraging and suggest that mathematical programming methods can be successfully applied to large-scale problems in delivery systems planning and to other



contexts of logistical system design. We feel it is important to structure the complete problem as a mathematical programming model. While we may apply heuristics to solve each of the modules, the mathematical programming framework permits us to delineate the approximations we are making in developing algorithms for each module.

We also believe that the use of a collection of linked subproblems, rather than a single model, is a good modeling approach. This approach is advantageous, not only because it reduces computational costs, but also because it eases the effort required to maintain the model. Though the speed and power of computers have increased dramatically in recent years, we feel that comprehensive monolithic models may not provide the flexibility necessary to evolve gracefully as the operations and needs of the user change.

## References

- [1] Assad A., "Models for Rail Transportation", *Transportation Research* **14A** (1980) 205-220.
- [2] Balakrishnan A. and S. C. Graves, "A Composite Algorithm for the Concave-Cost LTL Consolidation Problem", WP#1669-85, Sloan School of Management, MIT( Boston, Mass, 1985).
- [3] Balakrishnan A., T.L. Magnanti and R.T. Wong, "A Dual-Based Procedure for Large-Scale Uncapacitated Network Design", Working Paper OR 161-87, Operations Research Center, MIT( Cambridge, MA, 1987). (To appear in *Operations Research*.)
- [4] Balinski M., "Fixed-cost Transportation Problems", *Naval Logistics Research Quarterly* **8** (1961) 41-54.
- [5] Bodin L., B. Golden, A. Assad and M. Ball, "Routing and Scheduling of Vehicles and Crews: The State of the Art", *Computers and Operations Research* **10** (1983) 67-211.
- [6] Bodin L., B. Golden, A. Schuster and W. Romig, "A Model for the Blocking of Trains", *Transportation Research* **14B** (1980) 115-120.
- [7] Bell W. J., L. M. Dalberto, M. L. Fisher, P. Kedia, R. G. Mack and P. J. Prutzman, "Improving the Distribution of Industrial Gases with an Online Computerized Routing and Scheduling Optimizer", *Interfaces* **13** (1983) 4-23.
- [8] Burns L. D., R. W. Hall, D. E. Blumfield and C.F. Daganzo, "Distribution Strategies that Minimize Transportation and Inventory Costs", *Operations Research* **33** (1985) 469-490.
- [9] Daganzo C.F., "The Break-Bulk Role of Terminals in Many-to-Many Logistics Networks", *Operations Research* **35** (1987) 543-555.

- [10] Cooper L. and C. Drebes, "An Approximate Solution Method for the Fixed Charge Problem", *Naval Research Logistics Quarterly* **14** (1967) 101-114.
- [11] Dijkstra E.W., "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik* **1** (1959) 269-271.
- [12] Dejax P.J. and T.G. Crainic, "A Review of Empty Flows and Fleet Management Models in Freight Transportation", *Transportation Science* **21** (1987) 227-247.
- [13] Eilon S., C. D. T. Watson-Gandy and N. Christofides, *Distribution Management* (Hafner Publishing Co., New York, NY, 1971) .
- [14] Fejes Toth L., *Lagerungen in der Ebene, auf der Kugel and im Raum* (Berlin-Gottingen-Heidelberg, 1953) .
- [15] Fisher M. L., "The Lagrangian Relaxation Method for Solving Integer Programming Problems", *Management Science* **27** (1981) 1-18.
- [16] Fisher M. L. and R. Jaikumar, "A Generalized Assignment Heuristic for Vehicle Routing", *Networks* **11** (1981) 109-124.
- [17] Fisher M. L., R. Jaikumar and L. Van Wassenhove, "A Multiplier Adjustment Method for the Generalized Assignment Problem", Working Paper 81-07-06, Decision Sciences, University of Pennsylvania( Philadelphia, PA, 1980).
- [18] Geoffrion A. M., "A Guide to Computer-Assisted Methods for Distribution Systems Planning", *Sloan Management Review* (1975) Winter 1975.
- [19] Geoffrion A. M., "Lagrangean Relaxation for Integer Programming", *Mathematical Programming Study* **2** (1977) 82-114.

- [20] Haimovich M. and T.L. Magnanti, "Extremum Properties of Hexagonal Partitioning and the Uniform Distribution in Euclidean Location", Working Paper OR 132-85, Operations Research Center, MIT (Cambridge, MA, 1987). (To appear in *SIAM Journal on Discrete Mathematics*.)
- [21] Lamar B. W., Y. Sheffi and W. B. Powell, "Bounding Procedures for Fixed Charge, Multicommodity Network Design Problems", Working Paper, Dept. of Civil Engineering, MIT (Cambridge, MA, 1984).
- [22] Magnanti T. L., "Combinatorial Optimization and Vehicle Fleet Planning: Perspectives and Prospects", *Networks* **11** (1981) 179-214.
- [23] Magnanti T. L. and R. T. Wong, "Network Design and Transportation Planning: Models and Algorithms", *Transportation Science* **18** (1984) 1-55.
- [24] Papadimitriou C.H., "Worst Case and Probabilistic Analysis of a Geometric Location Problem", *SIAM Journal on Computing* **10** (1981) 542-557.
- [25] Perl J. and M. S. Daskin, "A Warehouse Location-Routing Problem", *Transportation Research* **19B** (1985) 381-396.
- [26] Powell W. B. and Y. Sheffi, "Interactive Optimization for Motor Carrier Load Planning", *Journal of Business Logistics* **7** (1986) 64-90.
- [27] Powell W. B. and Y. Sheffi, "Design and Implementation of an Interactive Optimization System for Network Design in the Motor Carrier Industry", *Operations Research* (1987) (To appear).
- [28] Shapiro J. F., "A Survey of Lagrangian Techniques for Discrete Optimization", *Annals of Discrete Mathematics* **5** (1979a) 113-138.

- [29] Shapiro J. F., *Mathematical Programming: Structure and Algorithms* (Wiley-Interscience, New York, 1979b) .
- [30] Singhal V., "Point-to-Point Package Delivery Systems", M.Sc. thesis, Operations Research Center, MIT ( 1984).
- [31] Transportation Policy Associates, *Transportation in America* (Washington D.C., 1983) .
- [32] "U.S. Industrial Outlook 1987", *U.S. Dept. of Commerce* (1987) .
- [33] Wong R. T., "Location and Network Design", in: O'hEigeartaigh M., Lenstra J K and Rinnooy Kan A H G, ed., *Combinatorial Optimization: Annotated Bibliographies* (Wiley-Interscience, New York, 1985) pp. 127-147.