ORSA Journal on Computing
Vol. 7, No. 1, Winter 1995

0899-1499/95/0701-0024 $01.25

The Distribution Problem with Carrier Service: A Dual Based Penalty Approach

MOUSTAPHA DIABY / *Department of Operations and Information Management, College of Business, University of Connecticut, Storrs, CT 06269; Email: DIABY@UCONNVM.UCONN.EDU*

R. RAMESH / *Department of Management Science & Systems, School of Management, State University of New York at Buffalo, Buffalo, NY 14260; Email: MGTRAMES@UBVMS.CC.BUFFALO.EDU*

(Received: June 1992; final revision received: October 1992; accepted: February 1994)

The distribution problem with carrier service is an important optimization problem that often arises in vehicle routing. The problem arises in the distribution of commodities from a central facility to a set of geographically distributed locations. Each location has a certain demand, the distribution vehicle has a load capacity, and the entire operation should be completed within a certain time. An outside carrier is available for direct service of locations from the central facility. The problem is to determine a feasible tour for the company vehicle and the locations to be served by the outside carrier such that the total cost of the operations is minimized. We model this problem and develop a solution technique using valid inequalities. A key feature of the solution approach is that it can easily be extended to solve similar problems with any type and number of side constraints as encountered in the current model. The algorithm has been extensively tested and computational results for problems having up to 200 locations are presented. The results show that the proposed approach is efficient and viable for solving problems of medium to large size. We also present a real-world application, and show how the model is implemented within the framework of an order scheduling and vehicle routing system. We present performance results with real-world data and demonstrate the operational efficiency achieved by using the proposed approach.

We consider a routing problem that arises in the distribution of commodities from a central facility in a company to a set of geographically distributed locations using a single vehicle. The problem is as follows. The vehicle starts from the facility and returns after the distribution. The vehicle has a finite load capacity, and each location has a known demand. The travel between locations involves both cost and time, and the entire distribution must be carried out within a certain time. In addition, an outside carrier who operates on an individual service contract basis is available. The carrier charges a certain amount separately for each location served, while assuring service within the stipulated time. The decision problem is to determine: (i) a set of locations to be served by the carrier, and (ii) a tour of the remaining locations to be served by the company vehicle starting and ending at the central facility, such that the total cost of the entire operation is minimized, subject to the load and time restrictions.

The above optimization problem is part of a larger distribution problem that we studied at an industrial chemical products company. We provide details on the operations of the company and discuss the model implementation in Section 4 of this paper. The larger problem involves several product types and a vehicle fleet. However, the volatility and reaction sensitivity of the chemicals require that the distribution of each class of products be carried out by a separate vehicle. Hence, the larger problem is decomposable into separable vehiclewise routing problems. The company uses an asymmetric costing mechanism in accounting travel costs. The travel cost is determined by: (i) vehicle recall considerations, (ii) costs of "deadheading" to an overnight residence area and the costs of accommodation and driver compensation, (iii) fuel, tariffs, insurance and depreciation costs, and (iv) a differential pricing mechanism used for the customers. The recall factors imply that if a move from location i to location j takes a vehicle farther away from the central facility, then it costs more than the move from j to i. The differential pricing is influenced by factors (i)–(iii) and also other factors such as business volume at a location and negotiated contract agreements.

The problem in consideration involves routing a vehicle through a subset of locations subject to constraints on load and time. We term this problem as the *Distribution Problem with Carrier Service* (DPCS). Problem DPCS is closely related to some of the generalizations of the traveling salesman problem (TSP), such as the orienteering problem (Tsiligirides[21]; Golden et al.[8]) of the prize-collecting traveling salesman problem (Balas[2]) and variants of TSP involving specific time windows such as those studied in Solomon,[20] Sexton and Bodin,[18] Psaraftis,[13] and Desrosiers et al.[6] In this research, we develop an optimal algorithm to solve problem DPCS using valid inequalities. A key feature of the solution approach is that it can easily be extended to solve similar problems with any type and number of general side constraints such as the time and load constraints encountered in the current model.

The organization of this paper is as follows. Section 1 presents the model structure and develops a bounding approach. Section 2 presents the solution methodology and

Subject classifications: Programming, integer, algorithms, branch and bound; Transportation, Models, Network, Vehicle Routing.

Section 3 presents the computational results with randomly generated problems. Section 4 discusses the implementation of the model and a performance assessment with real-world distribution data provided by the company where the study was conducted. The conclusions are presented in Section 5.

1. Problem Structure

Problem DPCS is specified as follows. Let $G = \{V, A\}$ represent a complete directed graph on n vertices, with vertex set $V = \{1, \ldots, n\}$ and arc set $A = \{i, j) \mid i, j = 1, \ldots, n\}$. Each vertex represents a location, and an arc represents the link between two locations. An arc (i, j) has a nonnegative cost c_{ij} and time t_{ij} associated with it. Each vertex j has a nonnegative service cost p_j associated with it. Vertex 1 represents the central facility, and $c_{ii} = t_{ii} = 0$ for all $i \in V$. Let D denote the vehicle load capacity, and b_j, $j = 2, \ldots, n$ denote the demands at the locations. Let $B = \sum_{j=2}^{n} b_j$ denote the total demand. Then, the load distributed using the outside carrier must be at least $(B - D)$. Let T denote the time within which the service is to be completed. Let $\{x_{ij} \mid i, j \in V\}$ denote the set of 0–1 decision variables. If $x_{jj} = 1$ in some solution, then location j is served by the outside carrier. The model is as follows.

$$\mathbf{P}: v(P) = \min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} + \sum_{j \in V} p_j x_{jj} \tag{1}$$

$$\text{s.t.} \quad \sum_{i \in V} \sum_{j \in V - \{i\}} t_{ij} x_{ij} \leqslant T \tag{2}$$

$$\sum_{i \in V} \sum_{j \in V - \{i\}} b_j x_{ij} \leqslant D \tag{3}$$

$$\sum_{j \in V} x_{ij} = 1 \quad \forall i \in V \tag{4}$$

$$\sum_{i \in V} x_{ij} = 1 \quad \forall j \in V \tag{5}$$

$$x_{11} = 0 \tag{6}$$

$$G(x) \text{ has exactly one cycle with more than one vertex} \tag{7}$$

$$x_{ij} = 0 \text{ or } 1 \quad \forall i, j \in V \tag{8}$$

In the above formulation $G(x)$ is a subgraph of G comprised of the vertex set V, with incidence vector x given by any feasible solution to problem P. Note that constraint (6) can easily be enforced by letting $p_1 = \infty$, and that we assume $b_1 = 0$. The load constraint (3) can also be equivalently expressed as: $\sum_{j \in V} b_j x_{jj} \geqslant (B - D)$. Note that any feasible solution to problem P is the union of the following subgraphs: (i) a tour of a set of vertices $V^* \subseteq V$ starting and ending at vertex 1 that is feasible to the time and load constraints, and (ii) a set of arcs $A^* = \{(j, j) \mid j \in V - V^*\}$. Each arc in A^* is a *self-loop* with only one vertex and represents a location served by the carrier. We investigate the polytopes associated with problem P in the following discussion.

Valid Inequalities

Initially, we distinguish between the following polyhedral sets that contain the polytope P: (i) $KP1 = \{x \in b^{n^2} \mid x$ satisfies (2)$\}$, (ii) $KP2 = \{x \in b^{n^2} \mid x$ satisfies (3)$\}$, (iii) $AP = \{x \in b^{n^2} \mid x$ satisfies (4) and (5)$\}$, (iv) $Q_1 = \{x \in b^{n^2} \mid x$ satisfies (2), (4) and (5)$\}$, (v) $Q_2 = \{x \in b^{n^2} \mid x$ satisfies (3), (4) and (5)$\}$, (vi) $Q_3 = \{x \in b^{n^2} \mid x$ satisfies (4), (5), (6) and (7)$\}$, where $n = |V|$ and b^n is the set of all 0–1 n-vectors. We denote the convex hull of problem (.) as *conv*(.) and its dimension as *dim*(.). KP1 and KP2 represent the knapsack polytopes from the time and load constraints, respectively. AP represents the assignment polytope. Q_1, Q_2, and Q_3 are sets of assignments satisfying the time, load, and subtour elimination constraints, respectively.

The structure of the 0–1 knapsack polytope has been well studied in the literature (see Balas[1]; Hammer et al.[9]). We refer to the valid inequalities for knapsack polytopes as cover inequalities. Consider $KP1$ for instance. $C \subseteq A$ is a *cover* for $KP1$ if $\sum_{(i,j) \in C} t_{ij} > T$. Moreover, if C is a cover and $\sum_{(i,j) \in C - \{(k,l)\}} t_{ij} \leqslant T$ $\forall (k, l) \in C$, then C is a *minimal cover* for $KP1$. Similarly, we can define a minimal cover C for $KP2$. Let $\Re_1$ and $\Re_2$ denote the sets of minimal covers for $KP1$ and $KP2$, respectively. Then, the following inequalities are valid for *conv*(P).

$$\sum_{(i,j) \in C} x_{ij} \leqslant |C| - 1, \quad \forall C \in \Re_1 \cup \Re_2 \tag{9}$$

Inequalities (9) can be strengthened using "lifting" or "extensions" of C (Nemhauser and Wolsey[12]). However, we only employ inequalities (9) because of their ease of implementation, which resulted in greater overall computational efficiency in solving problem P.

Consider polytopes Q_1 and Q_2. Note that $conv(P) \subseteq conv(Q_1) \subseteq conv(KP1)$ and $conv(P) \subseteq conv(Q_2) \subseteq conv(KP2)$. Consider a minimal cover $C \in \Re_1 \cup \Re_2$. If C satisfies constraints (4), (5), and $x_{ii} = 0$, $\forall i \in V$, then it is a full assignment in G, and the following inequality is valid for *conv*(P):

$$\sum_{(i,j) \in C} x_{ij} \leqslant |C| - 2. \tag{10}$$

This is because two distinct perfect matchings of a bipartite graph with equal cardinality vertex sets must differ in at least two matched arcs. Inequalities (10) are stronger than (9), and require no additional computational effort. Consider another type of inequality arising out of the time constraint as follows. Let $r_i = \min\{t_{ij}, \forall j \in V\}$ and $q_j = \min\{t_{ij}, \forall i \in V\}$. Let V_r, $V_q \subseteq V$ be such that $\sum_{i \in V_r} r_i > T$ and $\sum_{j \in V_q} q_j > T$. The following inequalities are clearly valid for *conv*(Q_1):

$$\sum_{i \in V_r} x_{ii} \geqslant 1 \tag{11}$$

$$\sum_{i \in V_q} x_{ii} \geqslant 1 \tag{12}$$

Similarly, let $U \subseteq V$ such that $\sum_{j \in V} b_j > D$. The following inequality is clearly valid for *conv*(Q_2):

$$\sum_{j \in U} x_{jj} \geqslant 1 \tag{13}$$

Consider polytope Q_3. This polytope also occurs in the prize-collecting traveling salesman problem (PCTSP) (Balas[2]). Consider any $s \subseteq V - \{1\}$ with $|S| \geqslant 2$. The following inequalities have been shown to be facet-defining for $conv(Q_3)$, provided $|S| \leqslant n - 2$:

$$\sum_{i \in S} \sum_{j \in S-\{i\}} x_{ij} + \sum_{i \in S-\{k\}} x_{ii} \leqslant |S| - 1 \quad \forall k \in S \tag{14}$$

Clearly, each inequality in (14) can be equivalently written as

$$\sum_{i \in S} \sum_{j \in S-\{i\}} x_{ji} + \sum_{i \in S-\{k\}} x_{ii} \leqslant |S| - 1 \quad \forall k \in S \tag{15}$$

Rewriting (5) as $\sum_{j \in V} x_{ji} = 1 \ \forall i \in V$ and summing over all i gives:

$$\sum_{i \in S} \sum_{j \in S-\{i\}} x_{ji} + \sum_{i \in S} \sum_{j \in V-S} x_{ji} + \sum_{i \in S} x_{ii} = |S| \tag{16}$$

Subtracting (16) from (15) gives a set of inequalities equivalent to (14) as follows:

$$\sum_{i \in S} \sum_{j \in V-S} x_{ij} + x_{kk} \geqslant 1 \quad \forall k \in S \tag{17}$$

Inequalities (14) and (17) have also been shown to be facet-defining for $conv(PCTSP)$ (Balas[2]). This is because, PCTSP has a special structure. The constraint set of PCTSP includes constraints (4)–(8) and an additional constraint $\sum_{i \in V} w_i x_{ii} \leqslant U$. PCTSP is a generalization of TSP, where every tour of a set $S \subseteq V$, $\{1\} \in S$ that satisfies the additional constraint is feasible. Consequently, every TSP tour is also feasible to PCTSP, and $conv(TSP) \subseteq conv(PCTSP) \subseteq conv(Q_3)$. Furthermore, (14) and (17) with $x_{ii} = 0$, $\forall i \in V$, have been known to be facet-defining for TSP (Lawler et al.[10]). Using a set of affinely independent points from the TSP facets and the special structure of the side constraint in PCTSP, the dimension of $conv(PCTSP)$ has been exactly determined and (14) and (17) have been shown to define its facets (Balas[2]).

In contrast to PCTSP, $conv(TSP) \not\subset conv(P)$ in general. The reason for this is that the side constraints (2) and (3) have no special structure. In fact, $conv(P) \cap conv(TSP) = \emptyset$ when constraint (3) is not redundant. Similarly, if constraint (2) is sufficiently tight, then no TSP tour will be feasible to problem P. Hence, facets of TSP do not naturally extend to facets of problem P. Moreover, the cardinality of the *equality set* of P depends on the problem instance. Note that the equality set of P consists of the explicitly stated equations (4), (5), and (6), plus equations that are implicit for the overall polytope. For example, consider $k \in V$: (i) if $b_k > D$ then $x_{kk} = 1$ in every feasible solution to problem P. Similarly, (ii) if $(t_{ik} + t_{kj}) > T$, $\forall i,\ j \in V,\ i \neq j$, and $(t_{1k} + t_{k1}) > T$, then $x_{kk} = 1$ in any feasible solution to problem P. Also, (iii) $x_{kj} = 0$ if $(t_{ik} + t_{kj} + t_{jt}) > T$, $\forall i,\ t \in V,\ i \neq t$ with $i,\ t \neq k,\ j$, and $(t_{1k} + t_{kj} + t_{j1}) > T$; and so on. Note that conditions (i)–(iii) can be applied recursively and in conjunction with one another to identify additional implicit equations of P and that other logical conditions for doing the same may exist as well. Letting ω denote the set of all implicit equations of P, we have $dim(P) \leqslant (n-1)^2 - |\omega|$, since there are n^2 variables and the rank of the explicit equations (4) and (5) is $(2n - 1)$. Clearly, ω cannot be exactly determined in general. Hence, the dimension of P cannot be exactly established in theory. Despite this, however, we show in the following proposition that the valid inequalities derived for Q_3 in particular are strong inequalities for problem P.

Proposition 1. *Every feasible solution to problem P belongs to some facet of polytope* Q_3.

Proof. Consider any feasible solution x to problem P. Let V^* be the corresponding set of vertices in the cycle that includes vertex 1, and A^*, the corresponding set of self-loop arcs. Let S be any connected subset of V^* or any subset of $V - V^*$. Clearly, inequality (14) for any $k \in S$ will be satisfied as an equality by x. ■

Proposition 2. *Let* $F = \{x \in P : \sum_{i \in S} \sum_{j \in S-\{i\}} x_{ij} + \sum_{i \in S-\{k\}} x_{ii} = |S| - 1$ *for some* $k \in S\}$ *for some* $S \subseteq V$. *Then, every* $x \in F$ *belongs to some facet of* P.

Proof. First, note that since (14) is a valid inequality for P, F is a supporting face of P. Now, the proposition follows from the fact that F is strictly included in P, and that every proper supporting face of P is the intersection of facets of P. ■

The above propositions imply that a point on a facet of Q_3 is also a point on a facet of P provided it is feasible to the side constraints. Hence, despite the lack of structure in (2) and (3), inequalities (14) (and therefore inequalities (17), since they are equivalent) can be used to generate points on the facets of P.

The Penalty Problem

Problem P is solved using valid inequalities (9)–(17) according to a penalty approach. Let Γ denote the set of all valid inequalities for P. In general, Γ can be expressed as: $\sum_{i \in V} \sum_{j \in V} a^k_{ij} x_{ij} \geqslant a^k_0$, $\forall k \in \Gamma$. Let π denote the matrix of objective function coefficients, where $\pi_{ij} = c_{ij}$ if $i \neq j$ and $\pi_{jj} = p_j$, $\forall j \in V$. Using these, we restate Problem P as follows:

$$\text{L: } v(L) = \min \sum_{i \in V} \sum_{j \in V} \pi_{ij} x_{ij} \tag{18}$$

$$\text{s.t.} \quad \sum_{j \in V} x_{ij} = 1 \quad \forall i \in V \tag{19}$$

$$\sum_{i \in V} x_{ij} = 1 \quad \forall j \in V \tag{20}$$

$$\sum_{i \in V} \sum_{j \in V} a^k_{ij} x_{ij} \geqslant a^k_0 \quad \forall k \in \Gamma \tag{21}$$

$$x_{ij} = 0 \text{ or } 1 \quad \forall i, j \in V \tag{22}$$

Let M denote the dual of the linear programming relaxation of L. Let $u_i\ (i \in V)$, $v_j\ (j \in V)$ and $\lambda_k\ (k \in \Gamma)$ denote the dual variables associated with (19), (20), and (21), respectively. Problem L with only the constraints (19), (20), and (22) constitutes an assignment problem (AP). Let $\bar{x}$

denote the optimal AP solution and $(\bar{u}, \bar{v})$ the corresponding dual variables. Then, from duality theory we have: $\bar{u}_i + \bar{v}_j = \pi_{ij}$ if $\bar{x}_{ij} = 1$, $\bar{u}_i + \bar{v}_j \leqslant \pi_{ij}$ if $\bar{x}_{ij} = 0$, and $v(AP) = \sum_{i \in V} \bar{u}_i + \sum_{j \in V} \bar{v}_j \leqslant v(\bar{L}) = v(\bar{M}) \leqslant v(L) = v(P)$, where $(\bar{.})$ denotes the linear programming relaxation of Problem (.).

Although problem $\bar{L}$ (and hence, its dual (M)) provides stronger bounds than AP, it cannot be solved directly, because Γ has an exponentially large number of constraints. Hence, we proceed by first solving problem AP and identifying the set of inequalities $(\bar{\Gamma})$ violated by its solution $\bar{x}$. Using $\bar{\Gamma}$ we set up a penalty problem to compute the penalties for the violation of $\bar{\Gamma}$ as follows.

$$\textbf{PP}: v(PP) = \max \sum_{k \in \bar{\Gamma}} \left(a_0^k - \sum_{i \in V} \sum_{j \in V} a_{ij}^k \bar{x}_{ij} \right) \lambda_k \tag{23}$$

$$\text{s.t.} \quad u_i + v_j + \sum_{k \in \bar{\Gamma}} a_{ij}^k \lambda_k = \pi_{ij} \text{ if } \bar{x}_{ij} = 1 \tag{24}$$

$$u_i + v_j + \sum_{k \in \bar{\Gamma}} a_{ij}^k \lambda_k \leqslant \pi_{ij} \text{ if } \bar{x}_{ij} = 0 \tag{25}$$

$$u_i, v_j \text{ unrestricted} \quad \forall i, j \in V \tag{26}$$

$$\lambda_k \geqslant 0 \quad \forall k \in \bar{\Gamma} \tag{27}$$

Note that any solution feasible to Problem PP is also feasible to Problem M. Let $M(\bar{\Gamma})$ denote Problem M with only $\bar{\Gamma}$ included. We establish a bound for Problem P using the penalty problem in the following theorem.

Theorem 1.

$$v(AP) \leqslant (v(AP) + v(PP)) \leqslant v(M(\bar{\Gamma})) \leqslant v(M) \leqslant v(L) = v(P).$$

Proof. Since $\bar{x}$ violates every inequality in $\bar{\Gamma}$, we have

$$a_0^k - \sum_{i \in V} \sum_{j \in V} a_{ij}^k \bar{x}_{ij} > 0, \quad \forall k \in \bar{\Gamma}.$$

This establishes the first inequality of the theorem. Now, denote the optimal solution to Problem PP by $(\hat{\lambda}, \hat{u}, \hat{v})$. Then, we have:

$$\hat{u}_i + \hat{v}_j = \pi_{ij} - \sum_{k \in \bar{\Gamma}} a_{ij}^k \hat{\lambda}_k, \quad \forall (i, j) \in V^2 \mid \bar{x}_{ij} = 1 \tag{28}$$

$$= \bar{u}_i + \bar{v}_j - \sum_{k \in \bar{\Gamma}} a_{ij}^k \hat{\lambda}_k, \quad \forall (i, j) \in V^2 \mid \bar{x}_{ij} = 1 \tag{29}$$

For any $i \in V$, let $j(i)$ be such that $\bar{x}_{i,j(i)} = 1$. Then, we have

$$\sum_{i \in V} \hat{u}_i + \sum_{j \in V} \hat{v}_j = \sum_{i \in V} (\hat{u}_i + \hat{v}_{j(i)}) \tag{30}$$

$$= \sum_{i \in V} \left\{ \bar{u}_i + \bar{v}_{j(i)} - \sum_{k \in \bar{\Gamma}} a_{i,j(i)}^k \hat{\lambda}_k \right\} \quad \text{[using (29)]} \tag{31}$$

$$= \sum_{i \in V} \bar{u}_i + \sum_{j \in V} \bar{v}_j - \sum_{k \in \bar{\Gamma}} \left\{ \sum_{(i,j) \in V^2 \mid \bar{x}_{ij} = 1} a_{ij}^k \hat{\lambda}_k \right\} \tag{32}$$

Hence,

$$v(AP) + v(PP) = v(AP) + \sum_{k \in \bar{\Gamma}} \left(a_0^k - \sum_{i \in V} \sum_{j \in V} a_{ij}^k \bar{x}_{ij} \right) \hat{\lambda}_k \tag{33}$$

$$= \sum_{i \in V} \hat{u}_i + \sum_{j \in V} \hat{v}_j + \sum_{k \in \bar{\Gamma}} a_0^k \hat{\lambda}_k \quad \text{[using (32)]} \tag{34}$$

This establishes the second inequality in the theorem. The third inequality follows from the fact that $\bar{\Gamma}$ is a subset of Γ. ■

2. Solution Methodology

Problem P is solved using a depth-first, branch-and-bound search strategy. The solution approach is as follows. Initially, the variables x_{ii} and x_{ij} that satisfy the conditions for their elimination described earlier are fixed using a recursive procedure. This reduces the problem size. Next, the AP relaxation of problem P is solved. If the solution is feasible to P, then the algorithm is terminated. Otherwise, an upper bound is obtained using an adaptation of a heuristic for the orienteering problem developed in Ramesh and Brown.[14] Then, all the valid inequalities violated by the AP solution are identified. Two approaches are used to incorporate these into the penalty problem. In the first approach these inequalities are considered in sequence in the penalty problem. This approach is similar to that of Balas and Christofides[3] for the traveling salesman problem. In the second approach all the inequalities are considered simultaneously and the penalty problem is solved to yield the bound improvement. We describe both the approaches in the following discussion.

Bounding Approach 1

Consider the AP solution $\bar{x}$, and some valid inequality $\sum_{i \in V} \sum_{j \in V} a_{ij}^1 x_{ij} \geqslant a_0^1$ violated by $\bar{x}$. Consider the penalty problem with only this inequality. Let $(\hat{\lambda}_1, \hat{u}, \hat{v})$ denote its solution. This solution yields the penalty $\mu_1 = \hat{\lambda}_1(a_0^1 - \sum_{i \in V} \sum_{j \in V} a_{ij}^1 \bar{x}_{ij})$ for the violation, and the lower bound improves to $v(AP) + \mu_1$. Note that the solution $(\hat{\lambda}_1, \hat{u}, \hat{v})$ is also the solution to a Lagrangean relaxation of the valid inequality which restricts the choice of the dual variables to preserve the primal solution $\bar{x}$ intact. Consequently, the Lagrangean bound is also $v(AP) + \mu_1$. Although $\bar{x}$ is preserved, the dual variables (and hence, the reduced costs) of AP could be different from those of the Lagrangean problem, because of the Lagrangean component in the objective. Let $\bar{\pi}$ and $\hat{\pi}$ denote the reduced cost matrices before and after solving the penalty problem, respectively. Starting from $\hat{\pi}$, the penalty problem with the next inequality in $\bar{\Gamma}$ is solved and the corresponding bound improvement is obtained. Repeating this step with each inequality in $\bar{\Gamma}$ yielding a set of penalties μ_k, $\forall k \in \bar{\Gamma}$, the lower bound eventually improves to $v(AP) + \sum_{k \in \bar{\Gamma}} \mu_k$. Redenoting the reduced cost matrix that results after each penalty problem as $\bar{\pi}$, we keep track of the variables with zero reduced costs at every stage in terms of $\overline{G} = \{(i, j) \mid \overline{\pi}_{ij} = 0\}$, which is a spanning subgraph of G.

Although the penalty problem can be solved to optimality at every stage, this can be time-consuming. Instead, we employ certain specialized procedures for different types of inequalities that yield strong penalties efficiently. These procedures are as follows.

Consider the cover inequalities (9) and (10). Minimal covers for KP1 are identified by first sorting the basic components of $\bar{x}$ in a descending order of the corresponding t_{ij}, and then determining a sequence of variables starting from the top of the list such that their total time just exceeds T. Minimal covers for KP2 are generated as follows. First, each basic variable x_{ij} is associated with b_j, and the basic components are sorted in the descending order of the corresponding b_j. Then, minimal covers are identified by determining a sequence of variables starting from the top of the list such that the resulting total load just exceeds the vehicle capacity.

Consider a minimal cover $C \in \mathfrak{R}_1 \cup \mathfrak{R}_2$. The objective of the penalty problem with inequality (9) is $\{(1 - |C|) + \sum_{(i,j) \in C} \bar{x}_{ij}\}\mu_1 = \mu_1$. We search for a largest μ_1 by adjusting the dual values corresponding to the AP solution $\bar{x}$ in such a way that the constraints of the penalty problem are satisfied. The strategy is to choose a μ_1 and increase either u_i or v_j (but not both) of every arc in C by μ_1 in such a way that the reduced costs of the nonbasic variables affected by this adjustment remain nonpositive. We say that a basic arc $(i, j) \in C$ is *covered by row*, if u_i is increased, and *covered by column*, if v_j is increased. Let I and J denote the index sets on u_i and v_j thus chosen. The following procedure is used to determine μ_1.

Procedure 1.

Step 1. (Find row and column minima)
Let $(i_1, j_1), \ldots, (i_m, j_m)$ denote the set of arcs in C. Let $\alpha_{i_h} = \min\{|\overline{\pi}_{i_h j}|, j \neq j_h\}$, $\forall h = 1, \ldots, m$, and $\beta_{j_h} = \min\{|\overline{\pi}_{i j_h}|, i \neq i_h\}$, $\forall h = 1, \ldots, m$. If $\alpha_{i_h} = \beta_{j_h} = 0$ for some h, then the procedure for the cover being considered is terminated with no bound improvement. Otherwise, let $I \leftarrow \emptyset$, $J \leftarrow \emptyset$, $\mu_1 \leftarrow \infty$, and go to Step 2.

Step 2. (Determine μ_1, I, J)
Let $\phi_h = \max\{\alpha_{i_h}, \beta_{j_h}\}$, $\mu_1 = \min\{\mu_1, \phi_h\}$, $\forall h = 1, \ldots, m$. If $\phi_h = \alpha_{i_h}$, then $I \leftarrow I \cup \{i_h\}$, $\forall h = 1, \ldots, m$. Otherwise, $J \leftarrow J \cup \{j_h\}$, $\forall h = 1, \ldots, m$. If $I = \emptyset$, go to Step 4. If $J = \emptyset$, go to Step 5. Otherwise, let $\theta = \min\{(|\overline{\pi}_{ij}/2|, \forall i \in I, \forall j \in J\}$ and $\mu_1 = \min\{\mu_1, \theta\}$, and go to Step 3.

Step 3. (Update reduced costs; I and J nonempty)
Let $\overline{\pi}_{ij} \leftarrow \overline{\pi}_{ij} + 2\mu_1$, $\forall (i, j) \in (I, J)$. Let $\overline{\pi}_{ij} \leftarrow \overline{\pi}_{ij} + \mu_1$, $\forall j \in J$, $\forall i$ such that $(i, j) \in A - C$ and i does not belong to I. Let $\overline{\pi}_{ij} \leftarrow \overline{\pi}_{ij} + \mu_1$ $\forall i \in I$, $\forall j$ such that $(i, j) \in A - C$ and j does not belong to J. Stop.

Step 4. (Update reduced costs; I empty)
Let $\overline{\pi}_{ij} \leftarrow \overline{\pi}_{ij} + \mu_1$, $\forall j \in J$, $\forall i$ such that $(i, j) \in A - C$. Stop.

Step 5. (Update reduced costs; J empty)
Let $\overline{\pi}_{ij} \leftarrow \overline{\pi}_{ij} + \mu_1$, $\forall i \in I$, $\forall j$ such that $(i, j) \in A - C$. Stop.

After performing Procedure 1, the basic variables that have at least one zero element in their respective rows and columns are deleted from the sorted list. A new cover is obtained from the updated list as before, and Procedure 1 is repeated until either the list is empty or no more cover can be generated. If Γ_1 is the set of inequalities thus generated, and $\mu_1(i)$, $i = 1, \ldots, |\Gamma_1|$ are their respective penalties, then the lower bound after this step is $v(AP) + \sum_{i=1}^{|\Gamma_1|} \mu_1(i)$. If the first cover employed from a sorted list happens to be a full assignment, then clearly only one inequality can be derived from that list. In this case however, the bound may be strengthened by using (10). The bounding procedure remains the same, except that the penalty is $2\mu_1$.

Consider inequalities (11), (12), and (13). Initially, r_i, $\forall i \in V$ and q_j, $\forall j \in V$ are determined. Then, $\{r_i\}$, $\{q_j\}$, and $\{b_j\}$ are sorted in their respective descending sequences. If for any $i \in V$, $\bar{x}_{ii} = 1$, then its corresponding r_i, q_j, and b_j are dropped from their respective sequences. Without loss of generality, consider the sequence $\{r_i\}$. Starting from the top of the list, identify a subsequence V_r of values such that $\sum_{i \in V_r} r_i$ just exceeds T. Since $\bar{x}_{ii} = 0$, $\forall i \in V_r$, inequality (11) is violated by $\bar{x}$. The objective of the penalty problem with this inequality is $(1 - \sum_{i \in V_r} \bar{x}_{ii})\mu_2 = \mu_2$. The search for μ_2 does not require any adjustment to the dual variables, because constraint (24) is not affected by this inequality. The following procedure is used to determine μ_2.

Procedure 2.

Step 1. (Determine μ_2)
Let $\mu_2 = \min\{|\overline{\pi}_{ii}|, \forall i \in V_r\}$. If $\mu_2 = 0$, Stop. Otherwise, go to Step 2.

Step 2. (Update reduced costs)
Let $\overline{\pi}_{ii} \leftarrow \pi_{ii} + \mu_2$, $\forall i \in V_r$. Stop.

After applying the above procedure, a new set V_r is determined as in Procedure 1, and the procedure is repeated until no further improvement is possible. Procedure 2 is also used with the sets V_q and U according to the inequalities (12) and (13), respectively. If Γ_2 is the set of all inequalities thus generated, and $\mu_2(i)$, $i = 1, \ldots, |\Gamma_2|$ are their respective penalties, then the lower bound after this step is $v(AP) + \sum_{k=1}^{k=2} \sum_{i=1}^{|\Gamma_k|} \mu_k(i)$.

Consider the inequalities (14). Let $S \subseteq V - \{1\}$ with $2 \leqslant |S| \leqslant n - 2$ be such that the set of vertices in S constitutes a subtour in $\bar{x}$. Then, $\bar{x}$ violates all the inequalities in (14). The objective of the penalty problem with each of the inequalities in (14) is $\{1 - |S| + \sum_{i \in S} \sum_{j \in S - \{i\}} \bar{x}_{ij}\}\mu_3 = \mu_3$. Since basic variables are involved in these inequalities, dual variables adjustment is necessary in determining μ_3. The following procedure is used for this purpose.

Procedure 3.

Step 1. (Find row and column minima)
Let $(i_1, j_1), \ldots, (i_m, j_m)$ denote a tour of the vertex set S in $\bar{x}$, such that $j_h = i_{h+1}$, $\forall h = 1, \ldots, (m - 1)$,

and $j_m = i_1$. Let I and J denote index sets as before. Let $\alpha_{i_h} = \min\{|\bar{\pi}_{i_h j}|, j \in V - S\}, \forall h = 1, \dots, m$, and $\beta_{j_h} = \min\{|\bar{\pi}_{i j_h}|, j \in V - S\}, \forall h = 1, \dots, m$. If $\alpha_{i_h} = \beta_{j_h} = 0$ for some h, then the procedure for the subtour being considered is terminated with no bound improvement. Otherwise, let $I \leftarrow \emptyset$, $J \leftarrow \emptyset$, and go to step 2.

Step 2. (Determine ϕ, I, J)
Let $\phi_h = \max\{\alpha_{i_h}, \beta_{j_h}\}, \forall h = 1, \dots, m$. If $\phi_h = \alpha_{i_h}$, then $I \leftarrow I \cup \{i_h\}$, else $J \leftarrow J \cup \{j_h\}, \forall h = 1, \dots, m$. Let $\mu_3 = \min\{\phi_h, h = 1, \dots, m\}$. Go to step 3.

Step 3. ($(i, j) \in \{S, S\}, i \in I, j \in J, i \neq j$)
Let $\theta_1 = \min\{|\bar{\pi}_{ij}|, (i, j) \in \{I, J\}, i \neq j\}$. Go to step 4.

Step 4. ($(i, j) \in \{S, S\}, i = j$)
Determine $\theta_2(k), \forall k \in S$ as follows. Consider arc $(k, k) \in \{S, S\}$. $\theta_2(k) = \infty$ if $k \notin I$ and $k \notin J$; $\theta_2(k) = |\bar{\pi}_{kk}|$ if either $k \in I$ or $k \in J$ but not both; $\theta_2(k) = |\bar{\pi}_{kk}|/2$ if $k \in I$ and $k \in J$. Determine $\theta_3(k) = \min\{|\bar{\pi}_{ii}| \mid i \in S, i \neq k, i \in I, j \in J\}, \forall k \in S$. Let $\theta_4 = \max\{\min\{\theta_2(k), \theta_3(k)\}, k \in S\}$ and ρ denote the value of k that gives θ_4. Let $\mu_3 = \min\{\mu_3, \theta_1, \theta_4\}$. If $\mu_3 = 0$, then no bound improvement is possible. Otherwise, go to step 5.

Step 5. (Update Reduced Costs)
$\bar{\pi}_{ij} \leftarrow \bar{\pi}_{ij} + \mu_3, \forall (i, j) \notin \{S, S\}$ if either $i \in I$ or $j \in J$. $\bar{\pi}_{\rho\rho} \leftarrow \bar{\pi}_{\rho\rho} + \mu_3$ if $\rho \in I$ or $\rho \in J$, but not both. $\bar{\pi}_{\rho\rho} \leftarrow \bar{\pi}_{\rho\rho} + 2\mu_3$ if $\rho \in I$ and $\rho \in J$. $\bar{\pi}_{ii} \leftarrow \bar{\pi}_{ii} + \mu_3$, $\forall i \in S, i \neq \rho, i \in I \cap J$. Stop.

In the above procedure, steps 1 and 2 are analogous to those in Procedure 1. A cell $(i, j) \in \{S, S\}$, $i \neq j$ that is twice covered limits μ_3 by $|\bar{\pi}_{ij}|$. This is shown in Step 3. Note that the system (14) actually yields k constraints. However, a positive penalty can be derived from at most one of these. Hence, the inequality that yields the maximum penalty is used. This is shown in Step 4. In both Procedures 1 and 3, the search for a positive penalty is terminated if $\alpha_{i_h} = \beta_{j_h} = 0$ for some h. However, it is possible to find a positive penalty by allowing some reduced cost to become positive, and subsequently adjusting the dual variables to achieve feasibility in the penalty problem. We term this approach the *Stepping Stone Approach* and illustrate it using an example below.

Table I shows the reduced costs in $\bar{x}$. The reduced costs of the basic cells are shown highlighted. Consider the subtour of vertices 3, 4, and 5 in $\bar{x}$. Step 1 of Procedure 3 yields $\alpha_3 = 2$, $\alpha_4 = 0$, $\alpha_5 = 5$, and $\beta_3 = 4$, $\beta_4 = 3$, and $\beta_5 = 0$. Let cell (3, 4) be covered by row 3 and cell (5, 3) by row 5. Note that for cell (4, 5), both α_4 and β_5 are zero. Nonetheless, assume that (4, 5) is covered by column 5, and let u_3, u_5 and v_5 be increased by some positive δ. As a result, $\bar{\pi}_{25}$ increases to δ. To counter this, we decrease u_2 by δ. This leads to $\bar{\pi}_{21}$ becoming $-\delta$. This is countered by increasing v_1 by δ. Note that there are no other zeros in column 1, and cells (3, 1) and (5, 1) are twice covered. Consequently, if we choose $\delta = 1.5$, the penalty constraints for all cells $(i, j) \notin \{S, S\}$ are satisfied. Further, if we choose vertex 5 as ρ, then the penalty constraints for all cells $(i, j) \in \{S, S\}$ are also satisfied with $\delta = 1.5$. The reduced cost matrix after this adjustment is shown in Table II. In this method, we alternately increase v_j and decrease u_i in some sequence as determined by the basic cells outside $\{S, S\}$ until a positive penalty is obtained. Although the process is stopped at this stage, further exploration for better penalties is possible. In this case, whenever a feasible penalty is found, it is saved. The procedure will ultimately stop when it reverts to increasing or decreasing a dual variable that has already been increased or decreased.

Each subtour of $\bar{x}$ could be used as above for bound improvement. If Γ_3 is the set of such inequalities and $\mu_3(i)$, $i = 1, \dots, |\Gamma_3|$ are the respective penalties, then the lower bound after this step is $v(AP) + \sum_{k=1}^{k=3} \sum_{i=1}^{|\Gamma_k|} \mu_k(i)$.

Consider the inequalities (17) and the subtour of S as before. Let $K_1 = \{S, V - S\}$ and $K_2 = (V - S, S)$ denote the corresponding cutsets. These cutsets yield the following inequalities: $\sum_{(i,j) \in K_1} x_{ij} + x_{kk} \geqslant 1 \;\; \forall k \in S$ and $\sum_{(i,j) \in K_2} x_{ij} + x_{kk} \geqslant 1 \; \forall k \in S$. Without loss of generality, consider cutset K_1. The objective term of the penalty problem for each inequality from K_1 is $(1 - \sum_{(i,j) \in K_1} \bar{x}_{ij} - \bar{x}_{kk})\mu_4 = \mu_4$. Similarly, the objective term for each inequality from K_2 is also μ_4. Note that only nonbasic components of $\bar{x}$ are affected by these inequalities. Hence, no dual variable adjustment is necessary. The following procedure is used to determine the maximum total penalty from all the inequalities using cutsets K_1 and K_2.

Procedure 4.

Step 1. (Determine $\delta_1 - \delta_4$)
Let $\delta_1 = \min\{|\bar{\pi}_{ij}|, \forall (i, j) \in K_1\}$, $\delta_2 = \min\{|\bar{\pi}_{ij}|, \forall (i, j) \in K_2\}$, $\delta_3 = \sum_{k \in S} |\bar{\pi}_{kk}|$. Let $\eta \subseteq S$ be a maximal subset such that $|\bar{\pi}_{kk}| > 0, \forall k \in \eta$. Let $\delta_4 = \delta_1 + \delta_2$. If $\delta_4 = 0$ or $\delta_3 = 0$, then the procedure for the subtour being considered is terminated with no bound improvement. Otherwise, If $\delta_3 \leqslant \delta_4$, go to step 2. Else, go to step 3.

Table I. Reduced Costs Before the Update

Row/Col	(1)	(2)	(3)	(4)	(5)
(1)	−7	**0**	−4	−3	−5
(2)	**0**	−3	−6	−5	0
(3)	−3	−2	−8	**0**	−2
(4)	−3	0	0	−6	**0**
(5)	−5	−6	**0**	−3	−4

Table II. Reduced Costs After the Update

Row/Col	(1)	(2)	(3)	(4)	(5)
(1)	−5.5	**0**	−4	−3	−3.5
(2)	**0**	−4.5	−7.5	−6.5	0
(3)	0	−0.5	−8	**0**	−0.5
(4)	−1.5	0	0	−6	**0**
(5)	−3.5	−6	**0**	−3	−2.5

Step 2. *($\delta_3 \leqslant \delta_4$, Update)*
Let $\delta_5 = (\delta_4 - \delta_3)/2$, $\mu_4 \leftarrow \mu_4 + \delta_3$; If $(\delta_1 - \delta_5) \geqslant 0$ and $(\delta_2 - \delta_5) \geqslant 0$, then: $\bar{\pi}_{ij} \leftarrow \bar{\pi}_{ij} + (\delta_1 - \delta_5)\ \forall(i, j) \in K_1$; $\bar{\pi}_{ij} \leftarrow \pi_{ij} + (\delta_2 - \delta_5)\ \forall(i, j) \in K_2$. If $(\delta_1 - \delta_5) < 0$, then: $\bar{\pi}_{ij} \leftarrow \bar{\pi}_{ij} + \delta_3\ \ \forall(i, j) \in K_2$. If $(\delta_2 - \delta_5) < 0$, then: $\bar{\pi}_{ij} \leftarrow \bar{\pi}_{ij} + \delta_3\ \ \forall(i, j) \in K_1$. Set $\bar{\pi}_{kk} \leftarrow 0\ \forall k \in S$; Stop.

Step 3. *($\delta_3 > \delta_4$, Update)*
Let $\delta_5 = (\delta_3 - \delta_4)/|\eta|$. $\mu_4 \leftarrow \mu_4 + \delta_4$; $\bar{\pi}_{ij} \leftarrow \bar{\pi}_{ij} + \delta_1\ \forall(i, j) \in K_1$; $\bar{\pi}_{ij} \leftarrow \bar{\pi}_{ij} + \delta_2\ \forall(i, j) \in K_2$; $\bar{\pi}_{kk} \leftarrow -\delta_5\ \ \forall k \in \eta$; Stop.

The above procedure derives the total penalty from all the inequalities in (17) by considering them together. Note that each inequality in (17) involves only one $k \in S$. Hence, considering the sum of the diagonal elements is equivalent to applying the inequalities in sequence. The penalties derived from the inequalities are distributed in such a manner that the chances of some reduced cost increasing to zero are minimized, while deriving the maximum penalty from the inequalities. If Γ_4 is the set of inequalities from the subtours of $\bar{x}$, and $\mu_4(i)$, $i = 1, \ldots, |\Gamma_4|$ are their respective penalties, then the lower bound after this step is $v(AP) + \sum_{k=1}^{k=4}\sum_{i=1}^{|\Gamma_k|} \mu_k(i)$. At each application of Procedures 1–4, the reduced costs of some nonbasic variables may increase to zero, and these arcs are added to $\bar{G}$.

Let $\bar{G}$ consist of ϕ disjoint components, denoted as $G_i \subseteq \bar{G}$, $i = 1, \ldots, \phi$. Let component G_1 contain vertex 1. If G_1 has a vertex a such that $G_1 - \{a\}$ has more than one component, then vertex a is said to be an *articulation point* of G_1. Consider one such component with vertex set S, and $\{1\} \notin S$. Let $L_1\{S,\ V - (S \cup \{a\})\}$ and $L_2 = \{V - (S \cup \{a\}), S\}$ denote cutsets. Then, clearly any feasible solution should satisfy either $\sum_{(i,j)\in L_1 \cup L_2} x_{ij} \geqslant 1$ or $x_{kk} = 1\ \ \forall k \in S$. We derive the following set of valid inequalities from this disjunction: $\sum_{(i,j)\in L_1 \cup L_2} x_{ij} + x_{kk} \geqslant 1\ \forall k \in S$. Therefore, if $\bar{x}_{kk} = 0$ for some $k \in S$, then $\bar{x}$ violates the inequality derived from vertex k. The objective term of the penalty problem in this case is $(1 - \sum_{(i,j)\in L_1 \cup L_2} \bar{x}_{ij} - \bar{x}_{kk})\mu_5 = \mu_5$. The following bounding procedure is used to determine these penalties.

Procedure 5.

Step 1. *(Initialize)*
Let $\bar{S} = \{k \mid k \in S,\ \bar{x}_{kk} = 0\}$, $\mu_5 \leftarrow 0$.

Step 2. *(Determine ν, ρ)*
If $\bar{S} = \emptyset$, stop. Otherwise, let $\nu_1 = \min\{|\bar{\pi}_{ij}|,\ \forall(i, j) \in L_1 \cup L_2\}$ and $\nu_2 = \max\{|\bar{\pi}_{kk}|,\ \forall k \in S\}$. Let ρ denote the value of k that gives ν_2. Let $\nu = \min\{\nu_1, \nu_2\}$. If $\nu = 0$, Stop. Otherwise, go to Step 3.

Step 3. *(Update)*
$\bar{\pi}_{ij} \leftarrow \bar{\pi}_{ij} + \nu,\ \forall(i, j) \in L_1 \cup L_2$; $\bar{\pi}_{\rho\rho} \leftarrow \bar{\pi}_{\rho\rho} + \nu$; $\mu_5 \leftarrow \mu_5 + \nu$. If $\nu = \nu_1$ stop. Otherwise, let $\bar{S} \leftarrow \bar{S} - \{\rho\}$, and go to step 2.

Inequalities as above can be derived for each component of G_1 that does not include vertex 1, and for each articulation point. If Γ_5 denotes the set of such inequalities, and $\mu_5(i)$, $i = 1, \ldots, |\Gamma_5|$ are the respective penalties, then the lower bound after this step is $v(AP) + \sum_{k=1}^{k=5}\sum_{i=1}^{|\Gamma_k|} \mu_k(i)$. The above analysis can be used to derive valid inequalities from other components of $\bar{G}$ as well. For example, if $G_t = \{V_t,\ A_t\}$, $G_t \subset \bar{G}$, $t \neq 1$ is any other component with cutsets $L_3 = \{V_t,\ V - V_t\}$ and $L_4 = \{V - V_t,\ V_t\}$, then clearly the inequalities $\sum_{(i,j)\in L_3 \cup L_4} x_{ij} + x_{kk} \geqslant 1$, $\forall k \in V_t$ are valid. The penalties from these inequalities can be derived using Procedure 5.

Bounding Approach 2

In this approach, the penalty problem with all the inequalities violated by $\bar{x}$ is solved as a simple linear programming problem using a simplex method modified to handle unrestricted variables. Since most of the variables are unrestricted, the problem is relatively easy to solve. The lower bound in this case is $v(AP) + v(PP)$. The reduced cost matrix $\bar{\pi}$ is then recomputed using the dual variables obtained from the penalty problem. Note that $\bar{x}$ is still preserved. The graph $\bar{G}$ is obtained from $\bar{\pi}$, and the inequalities from the components of $\bar{G}$ (if any) are employed to strengthen the lower bound using Procedure 5.

Finding An Embedded Solution

The graph $\bar{G}$ could contain an embedded feasible solution to problem P, that is also an alternate optimal solution to a Lagrangean relaxation of the valid inequalities. If such a solution is found, then it is tested for complementary slackness. These conditions require the incidence vector of such a solution to satisfy all the inequalities that yielded positive penalties as strict equalities. The search for an embedded solution is conducted as follows. We keep track of the components of $\bar{G}$ throughout the bounding process. Initially, each cycle in $\bar{x}$ is a component, and is easy to identify. When nonbasic arcs are added to $\bar{G}$, some of the components may become connected, and we keep track of them. At the end of the bounding step, $\bar{G}$ could either have a single component or several components. If it has several, then it should also contain arcs (i, i), $\forall i \notin G_1$ for a feasible solution to P to exist. If this condition is not satisfied, then the search for a feasible solution is terminated. If either this condition is satisfied or $\bar{G}$ has only one component, then G_1 is searched for a feasible solution. G_1 contains a feasible solution if there exists a $G^* \subseteq G_1$ that is a tour including vertex 1 and satisfying (2) and (3), and $(i, i) \in \bar{G}$, $\forall i \notin G^*$. Although finding a feasible solution can be as difficult as finding the optimal tour in terms of worst-case behavior, this problem is considerably easier in most graphs (see Christofides[5] for a related discussion on Hamiltonian cycles). We employ a specialized implicit enumeration procedure based on the method of Roberts and Flores[16] (also see Selby[17]) for this. The search is conducted within a time limit.

Separation and Branching

If $\bar{G}$ does not yield an optimal solution to problem P, then a branch-and-bound search is conducted. Let LB and UB denote current best lower and upper bounds, respectively. If $\bar{G}$ yields a feasible solution with objective value z^* such that $z^* \leqslant UB$, then UB is updated. The problem size is reduced by fixing the nonbasic variables x_{ij} that satisfy $\bar{\pi}_{ij} < 0$ and $(LB - \bar{\pi}_{ij}) \geqslant UB$ to zero. We experimented with several separation and branching strategies. These include adaptations of Smith et al.,[19] Bellmore and Malone,[4] and Garfinkel[7] for asymmetric TSP. These strategies provide disjunctions based on subtour elimination. We also adapted these strategies to a set of basic arcs in $\bar{x}$ that violate the time or load constraint. Our preliminary study showed that if these constraints allow no more than about 30% of the vertices to be included in the tour, then disjunctions based on either time or load constraints perform better than subtour disjunctions. Accordingly, we employed subtour disjunctions only if the incumbent solution at hand contained more than 30% of the vertices in our detailed study. The branching strategy of Smith et al.[19] performed the best with all types of disjunctions.

3. Computational Results

The proposed algorithm has been programmed in *FORTRAN 77* and implemented on *IBM*3081/*GX*. The algorithm has been extensively tested with randomly generated problems using a completely randomized block design study. The design consists of three parameters: n (problem size), α (time parameter), and β (load parameter). For each randomly generated problem, a TSP tour using an adaptation of the heuristic of Lin and Kernighan[11] with cost matrix $\{c_{ij}\}$ and $c_{ii} = \infty$, $\forall i \in V$ is determined. The TSP tour provides a way of serving all the locations by the company vehicle at a cost close to the minimum, if time and load constraints were absent. Let Δ denote the total time of this tour. If $T > \Delta$, then this is probably a good solution if the load constraint permits. Accordingly, we set $T = \alpha\Delta$, $0 < \alpha \leqslant 1$, where α denotes the fraction of the time required by the TSP solution. Similarly, we set $D = \beta B$, $0 < \beta \leqslant 1$, where β is the fraction of the total demand. In our experiments, we set α and β at two levels: (0.33, 0.67). The costs c_{ij} have been uniformly generated in the interval (0, 500) and p_j in the interval $\{(c_{1j} + c_{j1}), 2(c_{1j} + c_{j1})\}$ for each j. The times t_{ij} have been uniformly generated in the interval (0, 100). Five problems have been solved in each cell of this design, using both the bounding approaches. The results with approach 1 are summarized in Table III. The results with approach 2 are quite comparable, and no major difference has been observed.

The results show that the average number of candidate problems is relatively modest. The CPU time shows an increasing trend with problem size, as expected. In general, the CPU times are significantly greater when α/β is (33/67) and (67/33) than when it is (33/33) and (67/67). This has an intuitive explanation. Note that problem P involves the selection of a subset of vertices to be included in the tour. The number of such candidate subsets is combinatorial. For example, (33/67) and (67/33) could admit more combinations than (33/33), resulting in greater CPU time at these intermediate levels. Similarly, if we view the problem as excluding a subset of vertices from a tour, a corresponding explanation can be derived for the greater CPU time with (33/67) and (67/33) than with (67/67). The duality gap after the first node of the branch-and-bound tree has been found to range between 1.6% and 3.1% of $v(P)$ in the class of problems tested. In comparison, the gap with the AP solution $\bar{x}$ at the first node ranged between 21.9% and 35.2% of $v(P)$.

We also conducted another experiment to study the behavior of the algorithm with respect to the cost and time matrices. In particular, we were interested in studying the behavior in "almost symmetric" conditions. We generated the upper triangular matrices of $\{c_{ij}\}$ and $\{t_{ij}\}$ as usual, and the lower triangular matrices randomly in the following intervals: $c_{ji} \in [c_{ij}(1 - \sigma), c_{ij}(1 + \sigma)]$ and $t_{ji} \in [t_{ij}(1 - \sigma), t_{ij}(1 + \sigma)]$. We considered four levels of σ: 0.2, 0.15, 0.1 and 0.05. The average CPU times for five problems in each size at each level of σ for $\alpha/\beta = 33/67$ are shown in Table IV. Similar behavior has been observed with other levels of α/β as well. The CPU times show an increasing trend with the cost and time matrices approaching the symmetric case. The reasons for this could be as follows. When the cost matrix tends to be symmetric, each cycle in the AP solution tends to involve just two vertices. Hence, the number of subtours proliferate, requiring considerable effort at bound improvement. In spite of this disadvantage, the valid inequalities performed quite well in tightening the AP bound even under near-symmetric conditions.

Table III. Computational Results Using Bounding Approach 1†

n	α	β	Avg. No. of Candidate Problems	CPU Time†† Min.	Avg.	Max
30	0.33	0.33	6.2	3.7	8.4	12.8
	0.33	0.67	11.6	7.5	9.9	14.5
	0.67	0.33	13.5	7.8	10.6	16.9
	0.67	0.67	4.9	2.1	7.9	9.8
50	0.33	0.33	14.2	21.6	24.2	33.7
	0.33	0.67	33.5	46.9	57.4	72.5
	0.67	0.33	36.4	51.8	63.5	70.6
	0.67	0.67	17.6	18.5	26.6	40.3
100	0.33	0.33	16.6	77.7	91.5	112.8
	0.33	0.67	35.2	106.3	139.8	159.9
	0.67	0.33	46.9	123.4	151.6	172.5
	0.67	0.67	15.1	61.3	87.6	103.5
150	0.33	0.33	18.2	101.3	126.9	143.2
	0.33	0.67	37.9	161.6	184.7	199.8
	0.67	0.33	32.4	165.2	172.3	192.2
	0.67	0.67	20.5	96.5	138.2	161.6
200	0.33	0.33	21.0	100.8	149.2	193.4
	0.33	0.67	39.5	179.5	219.6	246.9
	0.67	0.33	36.7	187.9	202.3	249.8
	0.67	0.67	17.4	131.1	147.7	175.2

†Five problems are solved in each problem category.
††The CPU time is in CPU seconds of IBM 3081/GX.

Table IV. Sensitivity to Level of Asymmetry†

n	CPU Time††				
	Random	$\sigma = 0.2$	$\sigma = 0.15$	$\sigma = 0.1$	$\sigma = 0.05$
30	9.9	27.5	33.1	58.7	79.3
50	57.4	91.8	123.7	182.6	226.7
100	139.8	177.8	216.4	261.5	294.5
150	184.7	212.9	241.3	297.8	369.9
200	219.6	291.6	393.7	458.2	512.7

†Five problems are solved in each problem category.
††The CPU time is in CPU seconds of IBM 3081/GX.

4. Model Implementation

The proposed model has been extensively tested with actual routing data obtained from the industrial chemicals company which initiated this research. We are currently in the process of implementing the model within the framework of a decision support system for vehicle routing with appropriate user interfaces and database management facilities. In this section, we give a more detailed background on the company and its operations, and describe how model DPCS is used for actual decision making. We also give an assessment of the computational performance of the model using the data provided by the company.

Company Background

The company concerned is located in Western New York, and is a major distributor of industrial chemicals in the Western New York–Northern Pennsylvania tiers. The company serves about 1000 customers on open-accounts and contracts, and about 500–1000 customers on a call-in basis and bids. Most of the customers are located within a radius of about 200 miles from the company. The products supplied by the company can be broadly classified into two types: *hazardous* and *nonhazardous* chemicals. The hazardous chemicals require specialized storage and handling, and are governed by the regulations of the Department of Transportation (DOT). The DOT classifies hazardous chemicals into 21 categories, and the company distributes chemicals from seven of these categories (namely, *chlorine, oxygen, flammable gas, flammable liquid, flammable solid, oxidizer,* and *corrosive*) on a regular basis. Occasionally, the company receives requests for chemicals from other categories as well. In such cases, the chemicals are procured from the manufacturers or other distributors and shipped directly to the customer. In particular, chemicals from the categories of flammable liquid, flammable solid, oxidizer, and corrosive are distributed in large quantities, and constitute about 70% of the total volume of hazardous chemicals distributed by the company.

Customer orders arrive randomly, and are communicated through mail, fax, or telephone. Each order lists the items required, quantities, and the date by which they should be delivered. If a customer order contains items from more than one hazard class, the order is split up according to the hazard classes into suborders which are logged separately into a delivery file. Orders logged into the delivery file are processed according to due dates. Lead times till due dates normally vary between same-day-delivery (rush orders) to about 9 days. Due to the competitive nature of the business, the company strives to meet any customer request on time, especially for customers having contracts and open-accounts. If the time till due date is not sufficient to fit the current delivery schedule of a company vehicle, then the order concerned is delivered using an outside carrier. The delivery costs using the outside carrier are relatively high, and the prevailing feeling at the company when this study was undertaken was that the outside carrier was being overused, and that costs could be reduced by better utilization and routing of company vehicles.

The company owns and operates a fleet of 11 large tractor trucks and 3 half-size vans. The tractor trucks have a weight limit of 36,500 lbs each, and the half-size vans have a limit of 3,500 lbs each. The tractors are used for long distance transportation, while the vans are used for relatively local deliveries. The DOT regulations on the transportation of hazardous materials are quite stringent, and require that either a truck carry at most one class of hazardous chemicals, or that additional protective packaging be provided with a solid segregating buffer between chemicals of two different classes if they are loaded in adjacent compartments of a truck. Due to factors such as the cost of additional protections, space utilization in a truck and the large quantities of the chemicals distributed, the policy of the company is not to carry chemicals from two different classes in the same vehicle. However, a vehicle could carry some nonhazardous chemicals (which are distributed in relatively small quantities) with any of the hazard categories.

The hazardous chemicals are classified into two groups, based on the volume and frequency of transactions in the groups. Group I comprises chlorine, oxygen, and flammable gas, and group II comprises flammable liquid, flammable solid, oxidizer, and corrosive. The main focus of the current study is on group II chemicals, which account for a major share of the business volume. Deliveries are made Monday through Saturday of each week. One tractor is assigned to each class of chemicals in group I and two tractors are assigned to each class in group II. Each vehicle assigned to group II leaves the company facility at the beginning of every alternate day of the week and completes a trip in two working days. The trips of the two vehicles assigned to each class of chemicals in group II are scheduled in an interleaved manner as follows. The first vehicle leaves on Mondays, Wednesdays, and Fridays, and returns by the end of the day on Tuesdays, Thursdays, and Saturdays. The second vehicle leaves on Tuesdays, Thursdays, and Saturdays, and returns by the end of the day on Mondays, Wednesdays, and Fridays. As a result, a vehicle leaves the company facility each day of the week (except Sundays) for each class of chemicals in group II. Because of the strong emphasis put on customer service, the company uses a one-day safety lead time in scheduling its deliveries. For example, orders that are due on Tuesday are assigned to the "Monday" vehicle of that week; orders due on

Wednesday are assigned to the "Tuesday" vehicle, and so on. The high volumes of the shipments and the random arrival of customer orders usually restrict this lead time to about a day in practical order scheduling. Each vehicle is allowed to complete a trip in not more than 16 hours (8 hours per day) of working time. Overnight stay at the end of the first day of the trip occurs at either the place of the last delivery on the first day, or the place of the first delivery on the second day, or at some suitable location between the two places. Drivers are rotated on a monthly basis between all the company vehicles used.

We have developed an order scheduling and trip routing procedure for each class of chemicals in group II. This procedure uses a *rolling horizon* approach, in which the horizon is limited to two days. Hence, in making a decision for orders due by day i, we also consider orders due by day $(i + 1)$. The justification for this approach comes from the fact that the time and capacity constraints do not usually allow consideration of more than two days of orders at a time. The basic idea of this approach is to include some (or all) of the orders of day $(i + 1)$ in the delivery cycle of days $(i - 1)$ and i if there is enough slack capacity in the company vehicle, and if this does not increase the total delivery cost for orders of days i and $(i + 1)$. We refer to this approach as the *Route-and-Roll* strategy, and develop a scheduling algorithm using this strategy in the following discussion.

The Route-and-Roll Algorithm

Consider the vehicle leaving on Monday of a given week. Clearly, all the orders due by Tuesday should be delivered either by this vehicle or the outside carrier, since the safety lead time is one day. We schedule the trip for this vehicle on Sunday night as follows. Let ϕ and ψ denote the sets of orders due on Tuesday and Wednesday, respectively. We initially solve model DPCS for the Monday vehicle with the set ϕ, and for the Tuesday vehicle with the set ψ. Denote the two models as $DPCS(\phi)$ and $DPCS(\psi)$, respectively. Let $z_c(\phi)$ and $z_o(\phi)$ denote the costs of the deliveries by the company vehicle and the outside carrier, respectively, in the solution of $DPCS(\phi)$. Similarly, define $z_c(\psi)$ and $z_o(\psi)$ for ψ. Hence, given the sets of orders ϕ and ψ that are known on Sunday night, one possible solution is to develop independent delivery schedules for ϕ and ψ as above. We term this approach as *independent scheduling*, which results in a total cost of $\lambda = z_c(\phi) + z_o(\phi) + z_c(\psi) + z_o(\psi)$. However, it may be possible to achieve a better solution by pooling the two sets of orders in determining delivery schedules. We term this approach as *pooled scheduling*, and is as follows. Let $\phi_c \subseteq \phi$ and $\phi_o \subseteq \phi$ denote the sets of orders to be delivered by the company vehicle and the outside carrier in the solution of $DPCS(\phi)$, respectively. Note that if there is enough slack capacity on the Monday vehicle, then we may be able to "squeeze in" some of the Wednesday orders in its trip. In order to do this, we solve model DPCS for the order set $(\phi_c \cup \psi)$ with $p_j = \infty\ \forall j \in \phi_c$. We refer to this model as $DPCS(\phi_c \cup \psi)$. The above condition in $DPCS(\phi_c \cup \psi)$ enforces that all the orders in ϕ_c be delivered by the company vehicle, since the solution of $DPCS(\phi)$ has shown that this is feasible, and we wish to minimize the use of the outside carrier. Let $\psi^* \subseteq \psi$ denote the set of Wednesday orders that can thus be squeezed onto the trip of the Monday vehicle from the solution of $DPCS(\phi_c \cup \psi)$. Let $z_c(\phi_c \cup \psi)$ and $z_o(\phi_c \cup \psi)$ denote the associated costs of the company vehicle and the outside carrier, respectively, in the solution of $DPCS(\phi_c \cup \psi)$. If ψ^* is empty, then we stop the procedure, and simply schedule the deliveries of the orders in ϕ according to the solution of $DPCS(\phi)$. Otherwise, we solve the DPCS model for the Tuesday vehicle with the set of orders $\psi - \psi^*$. We refer to this model as $DPCS(\psi - \psi^*)$. Let $z_c(\psi - \psi^*)$ and $z_o(\psi - \psi^*)$ denote the associated costs of the company vehicle and the outside carrier, respectively, in the solution of $DPCS(\psi - \psi^*)$. Hence, the total cost of pooled scheduling is $\nu = \{z_o(\phi) + z_c(\phi_c \cup \psi) + z_c(\psi - \psi^*) + z_o(\psi - \psi^*)\}$. If $\nu \leqslant \lambda$, then we schedule delivery of the set $(\phi_c \cup \psi^*)$ by the Monday vehicle according to the solution of $DPCS(\phi_c \cup \psi)$, and the set ϕ_o by the outside carrier. Otherwise, we simply schedule the deliveries of orders in ϕ according to the solution of $DPCS(\phi)$.

In the above strategy, note that we make routing decisions on Sunday night only for the Tuesday orders. A reason for this is that the Wednesday orders are still not firm, as there can be cancellations, changes to orders, and arrival of new orders on Monday. However, we employ the information available on the Wednesday orders to make the best decisions for the Tuesday orders on Sunday night. The above sequence of steps is repeated on Monday night in scheduling the delivery of Wednesday orders (by which time, the Wednesday orders are firm). In this case, we consider the Wednesday and Thursday orders in scheduling the delivery of Wednesday orders. The scheduling horizon thus rolls in succession, by considering one additional day at each scheduling stage. We refer to this procedure as the *Route-and-Roll Algorithm*.

In the DPCS model employed in the Route-and-Roll algorithm, each order represents a vertex. Note that there can be several orders from the same customer. In this case, the travel cost between the corresponding vertices is zero, and the travel time is simply the unloading time for an order before the other. The key idea in the Route-and-Roll algorithm is to schedule a set of deliveries by looking a day ahead in optimizing the decisions. The algorithm is run every night, and is used for scheduling the next day trip. Although it is possible to model the above process by considering longer look-ahead periods or even the entire set of known orders at any time, such an approach is not practical because of the tightness of the capacity and time constraints, and the lack of certainty of the order set beyond a day.

We provide test results from a sample of one month data on the distribution of flammable liquids in Table V. We ran the Route-and-Roll algorithm for each working day of the month, and derived order schedules and route plans for the vehicles in each day. Table V provides the number of orders involved in the DPCS models for ϕ, ψ, $(\phi_c \cup \psi)$ and $(\psi - \psi^*)$ each day, and the respective CPU times. In the current context, a vehicle delivers on an average between

Table V. Performance Characteristics of the Route-and Roll Algorithm

Day	A		B		C		D		E	F	G
	CARD.	TIME	CARD.	TIME	CARD.	TIME	CARD.	TIME			
1	70	108.5	51	101.3	110	187.5	45	77.5	65	11	P
2	62	101.3	67	106.8	121	178.2	—	—	54	8	I
3	84	113.7	47	77.7	119	202.4	—	—	72	12	I
4	53	111.8	62	113.9	112	177.8	51	109.2	61	3	P
5	69	173.5	36	142.9	96	236.8	31	68.4	65	9	P
6	58	68.7	79	92.5	126	254.5	72	88.6	65	11	P
7	90	123.5	73	97.3	148	224.3	63	131.9	85	15	P
8	74	119.2	44	91.0	98	139.6	—	—	54	20	I
9	51	181.4	63	87.5	107	228.2	—	—	44	7	I
10	69	73.8	33	79.2	96	142.3	26	77.5	70	6	I
11	45	112.5	52	71.7	97	174.6	46	112.0	51	—	I
12	60	103.7	50	110.6	107	209.1	—	—	57	3	I
13	64	104.2	37	106.5	101	163.7	—	—	64	—	I
14	48	97.2	59	111.3	107	191.3	54	98.7	53	—	P
15	75	118.8	68	109.9	143	249.9	—	—	75	—	I
16	81	132.8	55	114.5	126	224.1	50	122.1	76	10	P
17	66	101.3	33	98.5	99	176.7	27	34.7	72	—	P
18	49	73.2	41	91.2	90	138.7	34	66.1	56	—	P
19	55	67.5	73	133.5	123	200.2	68	146.2	60	5	P
20	79	111.3	90	173.8	158	238.5	85	138.7	73	11	P
21	93	109.8	58	111.5	151	241.9	—	—	78	15	I
22	63	133.5	27	58.9	90	184.8	—	—	63	—	I
23	39	108.4	43	109.1	82	177.8	—	—	38	1	I
24	57	116.2	36	92.4	93	166.4	33	102.5	60	—	P
25	48	109.9	30	97.8	78	140.9	—	—	48	—	I

(A): $DPCS(\phi)$; (B): $DPCS(\psi)$; (C): $DPCS(\phi_c \cup \psi)$; (D): $DPCS(\psi - \psi^*)$; (E): Number of orders assigned to company vehicle; (F): Number of orders assigned to outside vehicle; (G): Choice of scheduling (Independent/Pooled); (CARD): Cardinality of vertex set; (TIME): CPU time in CPU seconds of IBM 3081/GX.

25 to 35 orders each day. Note that $DPCS(\psi - \psi^*)$ is applicable only for those days where ψ^* is nonempty. Table V also shows the number or orders scheduled for the company vehicle and the outside carrier, respectively, in the final schedule derived for each day. Finally, we also indicate in this table whether a schedule is derived from the independent or pooled strategies. We computed the total savings in the cost of distribution using the proposed approach over the current system used in the company. We obtained from the company the total cost actually incurred during the month over which we conducted this experiment, and compared it with the total cost incurred using the proposed approach. We found a savings of 21.6% in the cost actually incurred. These results illustrate the operational efficiency obtained from using the proposed model.

5. Conclusion

The distribution problem with carrier service is an important optimization problem that often arises in vehicle routing. The problem arises in the distribution of commodities from a central facility to a set of geographically distributed locations. The distribution is carried out by a company vehicle under time and load constraints, and an outside carrier is available for direct service of locations from the central facility. The problem is to determine a feasible tour for the company vehicle and the locations to be served by the outside carrier such that the total cost of the operations is minimized. We model this problem and develop a solution technique using valid inequalities. A key feature of the solution approach is that it can easily be extended to solve similar problems with any type and number of general side constraints such as the time and load constraints encountered in the current model. The algorithm has been implemented and extensively tested. We also present a real-world application, and show how the model is implemented within the framework of an order scheduling and vehicle routing system. We present performance results with real-world data and demonstrate the operational efficiency achieved by using the proposed approach. The computational requirements of the algorithm are reasonable for problems of medium to large size, and show that the proposed approach is efficient and viable.

References

1. E. BALAS, 1975. Facets of the Knapsack Polytope, *Mathematical Programming, 8,* 146–164.
2. E. BALAS, 1989. The Prize Collecting Traveling Salesman Problem, *Networks, 19,* 621–636.

3. E. BALAS and N. CHRISTOFIDES, 1981. A Restricted Lagrangean Approach to the Traveling Salesman Problem, *Mathematical Programming 21*, 19–46.
4. M. BELLMORE and J.C. MALONE, 1971. Pathology of Traveling Salesman Subtour Elimination Algorithms, *Operations Research 19*, 278–307.
5. N. CHRISTOFIDES, 1976. *Graph Theory: An Algorithmic Approach*, Academic Press, New York.
6. J. DESROSIERS, Y. DUMAS, and F. SOUMIS, 1986. A Dynamic Programming Solution of the Large-Scale Single-Vehicle Dial-a-Ride Problem with Time Windows, *American Journal of Mathematical and Management Sciences 6*, 301–326.
7. R.S. GARFINKEL, 1973. On Partitioning the Feasible Set in a Branch-and-Bound Algorithm for the Asymmetric Traveling Salesman Problem, *Operations Research 21*, 340–343.
8. B.L. GOLDEN, Q. WANG, and L. LIU, 1988. A Multifaceted Heuristic for the Orienteering Problem, *Naval Research Logistics 35*, 359–366.
9. P.L. HAMMER, E.L. JOHNSON, and U. PELED, 1975. Facets of Regular 0-1 Polytopes, *Mathematical Programming 8*, 179–206.
10. E.L. LAWLER, J.K. LENSTRA, A.H.G. RINNOOY KAN, and D. SHMOYS, 1985. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Wiley, New York.
11. S. LIN and B.W. KERNIGHAN, 1973. An Effective Heuristic Algorithm for the Traveling Salesman Problem, *Operations Research 21*, 498–516.
12. G.L. NEMHAUSER, and L.A. WOLSEY, 1988. *Integer and Combinatorial Optimization*, Wiley, New York.
13. H.N. PSARAFTIS, 1980. A Dynamic Programming Solution to the Single Vehicle Many-to-Many Immediate Request Dial-a-Ride Problem, *Transportation Science 14*, 130–154.
14. R. RAMESH and K.M. BROWN, 1991. An Efficient Four-Phase Heuristic for the Generalized Orienteering Problem, *Computers & Operations Research 18*, 151–165.
15. R. RAMESH, Y.S. YOON, and M.H. KARWAN, 1992. An Optimal Algorithm for the Orienteering Tour Problem, *ORSA Journal on Computing 4*, 155–165.
16. S.M. ROBERTS and B. FLORES, 1966. Systematic Generation of Hamiltonian Circuits, *Communications of the ACM 9*, 690–693.
17. G.R. SELBY, 1970. The Use of Topological Methods in Computer Aided Circuit Layout, Ph.D. Thesis, London University, U.K.
18. T. SEXTON and L. BODIN, 1985. Optimizing Single Vehicle Many-to-Many Operations with Desired Delivery Times: I and II, *Transportation Science 19*, 378–435.
19. T.H.C. SMITH, V. SRINIVASAN, and G.L. THOMPSON, 1977. Computational Performance of Three Subtour Elimination Algorithms for Solving Asymmetric Traveling Salesman Problems, *Annals of Discrete Mathematics 1*, 495–506.
20. M. SOLOMON, 1987. Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints, *Operations Research 35*, 254–267.
21. T. TSILIGIRIDES, 1984. Heuristic Methods Applied to Orienteering, *Journal of Operational Research Society 5*, 153–162.