Location-Routing Problems with Distance Constraints

Rosemary T. Berger
Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015, rtb3@lehigh.edu

Collette R. Coullard
School of Mathematics & Computer Science, Lake Superior State University, Sault Ste. Marie, MI 49783, ccoullard@lssu.edu

Mark S. Daskin
Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60208, m-daskin@northwestern.edu

Abstract

An important aspect of designing a distribution system is determining the locations of the facilities. For systems in which deliveries are made along multiple stop routes, the routing problem and the location problem must be considered simultaneously. In this paper, a set-partitioning-based formulation of an uncapacitated location-routing model with distance constraints is presented. An alternate set of constraints is identified that significantly reduces the total number of constraints and dramatically improves the linear programming relaxation bound. A branch-and-price algorithm is developed to solve instances of the model. The algorithm provides optimal solutions in reasonable computation time for problems involving as many as 10 candidate facilities and 100 customers with various distance constraints.

Key words: *location-routing; column generation; branch-and-price*

Introduction

Determining the locations of facilities within a distribution network is an important decision that impacts not only the profitability of an organization but the ability to serve customers. Classical assumptions in location modeling are that deliveries are made on out-and-back routes visiting a single customer (or that customers travel individually to the site). Under this assumption, the cost of delivery is independent of other deliveries made. In many contexts, however, deliveries are made along multiple stop routes visiting two or more customers; in this case, the cost of delivery depends on the other customers on the route and the sequence in which they are visited. In order to capture accurately the cost of multiple stop routes within a location model, the routing problem must be solved at the same time as the location problem.

In its most general form, the location-routing problem (LRP) seeks to minimize total cost by simultaneously selecting a subset of candidate facilities and constructing a set of delivery routes that satisfy the following constraints: (i) customer demands are satisfied without exceeding vehicle or facility capacities, (ii) the number of vehicles, the route lengths and the route durations do not exceed the specified limits and (iii) each route begins and ends at the same facility. In this paper, we consider an uncapacitated LRP with distance constraints, which is a version that arises in a number of application contexts.

1. *Perishable goods delivery* problems have temperature restrictions to prevent spoilage that often translate into route duration or route length constraints. For example, Gorr et al. (2001) report that hot meals delivered to home-bound seniors through programs such as Meals on Wheels must be at least 140° F. In their study, the temperature requirement for the last meal delivered on a route translated into a total route duration of 45 minutes. Therefore, the decisions of where to locate meal preparation kitchens and how to sequence customers on multiple stop routes must be made simultaneously to ensure that the delivered meals meet the 140° F temperature requirement.

2. *Time critical delivery* problems, such as express package delivery, have time deadline restrictions that limit the duration or length of routes. For example, Barnesandnoble.com is able to take orders as late as 11 a.m. on weekdays from Manhattan customers and deliver books, CDs and DVDs by 7 p.m. to businesses and residences (Paltrow, 2003). Often time critical delivery problems involve penalties if the deadline is not met. For example, FedEx offers a money-back guarantee for its services; in the event of a package being delivered after a published commitment time, FedEx will refund or credit the transportation charges. The decisions of where to locate crossdock locations and how to sequence customers on multiple stop routes must be made simultaneously to ensure that packages arrive at their final destinations prior to the time deadline.

Webb (1968) and Christofides and Eilon (1969) were among the first to recognize the error introduced into location modeling by the out-and-back representation of delivery costs. By the mid-1970s, models, solution procedures and applications of LRPs were beginning to appear in the literature. Laporte (1988) summarizes the literature published prior to 1988. More recently, Min et al. (1998) develop and use a hierarchical taxonomy and classification scheme based on problem characteristics and solution methodology to review the LRP literature. Most of the research to date has focused on heuristic methods since LRPs merge two NP-hard problems. The heuristics generally decompose the problem into its three components, facility location, customer allocation to facilities and vehicle routing, and solve a series of well-known problems such as p-median, location-allocation and vehicle routing.

Exact methods have been developed for a small number of LRP models that are derived from two-index flow formulations for the vehicle routing problem (VRP). Laporte and Nobert (1981) solve a single depot model by a constraint relaxation method. The method was inspired by the branch-and-bound algorithm for the TSP developed by Miliotis (1976). Laporte (1986) develops an equivalent model and also extends the model to the case where the number of vehicles used is a variable in the model. Laporte et al. (1983) solve a multi-depot problem in which at most p facilities are located by adapting Miliotis' (1978) REVERSE algorithm. The largest problems solved have seven candidate facilities and 40 customers. Laporte et al. (1986) solve a multi-depot capacitated LRP using a constraint relaxation method. In their work, the largest problem solved to optimality has eight candidate facilities and 20 customers. Laporte et al. (1988) use a branch-and-bound procedure to solve asymmetric LRPs that include as many as three candidate facilities and 80 customers.

Success in developing exact methods for solving larger instances of LRPs is likely to come from leveraging the advances in exact methods for solving VRPs and other difficult combinatorial optimization problems. Motivated by the success of set partitioning formulations for a variety of transportation problems, such as the VRP with time windows (e.g. Desrosiers et al., 1984), the pickup and delivery problem with time windows (e.g. Savelsbergh and Sol, 1998) and the crew scheduling problem (e.g. Hoffman and Padberg, 1993), we investigate the effectiveness of set partitioning formulations and branch-and-price algorithms in the context of developing exact algorithms for LRPs.

There are two main contributions of this paper. First, we present a new formulation for the LRP with distance constraints and we identify an alternative set of constraints that dramatically improves the linear programming (LP) relaxation bound. Second, we develop a branch-and-price algorithm for the LRP with distance constraints. The pricing problem, which decomposes into a set of elementary shortest path problems with a single resource constraint, is solved using the extended label correcting algorithm of Feillet et al. (2004). In §3, we show that our algorithm can solve optimally instances as large as 10 candidate facilities and 100 customers with various distance constraints.

The remainder of the paper is organized as follows. In §1, we present the set-partitioning-based formulation of the LRP with distance constraints, and we develop valid inequalities that strengthen the LP relaxation bound. In §2, we describe the details of the branch-and-price algorithm including the solution of the pricing subproblem. We discuss the computational experiments in §3 and conclude with some directions for future research in §4.

1 Problem Formulation

In this section, we present a new set-partitioning-based formulation of the LRP with distance constraints. The objective of the LRP with distance constraints is to select a set of locations and to construct a set of associated delivery routes in such a way as to minimize facility costs plus routing costs. The set of routes must be such that each customer is visited exactly once by one route and that the length of each route does not exceed the maximum distance.

1.1 Initial Model

Let I be the set of customer locations and J be the set of candidate facility locations. We define the graph $G = (N, A)$, where $N = I \cup J$ is the set of nodes and $A = N \times N$ is the set of arcs. We let d_{ij} for all $(i, j) \in A$ be the distance between nodes i and j; the distances satisfy the triangle inequality. For applications in which the distance constraint applies to the length of the route to the last customer instead of the length of the return trip to the depot, we set d_{ij} to 0 for all (i, j) with $i \in I$ and $j \in J$.

We define a feasible route k associated with facility j as a simple circuit that begins at facility j, visits one or more customer nodes and returns to facility j and that has a total distance of at most the maximum distance, denoted M. Then, we let P_j denote the set of all feasible routes associated with the facility j for all $j \in J$. The cost of a route $k \in P_j$ is the sum of the costs of the arcs in the route. The cost of an arc $(i, j) \in A$ is proportional to the distance d_{ij} to reflect distance related operating costs.

Parameters

$$a_{ijk} = \begin{cases} 1 & \text{if route } k \text{ associated with facility } j \text{ visits customer } i, \forall i \in I, \forall j \in J, \forall k \in P_j \\ 0 & \text{otherwise} \end{cases}$$

$$c_{jk} = \text{cost of route } k \text{ associated with facility } j, \forall j \in J, \forall k \in P_j$$

$$f_j = \text{fixed cost associated with selecting facility } j, \forall j \in J$$

$$\alpha = \text{objective weighting factor}$$

Decision Variables

$$X_j = \begin{cases} 1 & \text{if facility } j \text{ is selected}, \forall j \in J \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{jk} = \begin{cases} 1 & \text{if route } k \text{ associated with facility } j \text{ is selected}, \forall j \in J, \forall k \in P_j \\ 0 & \text{otherwise} \end{cases}$$

$$\text{(LRP-DC) Minimize} \quad \alpha \cdot \sum_{j \in J} f_j X_j + \sum_{j \in J} \sum_{k \in P_j} c_{jk} Y_{jk} \tag{1}$$

$$\text{subject to} \quad \sum_{j \in J} \sum_{k \in P_j} a_{ijk} Y_{jk} = 1 \qquad \forall i \in I \tag{2}$$

$$X_j - Y_{jk} \geq 0 \qquad \forall j \in J, \forall k \in P_j \tag{3}$$

$$X_j \in \{0,1\} \quad \forall j \in J \tag{4}$$

$$Y_{jk} \in \{0,1\} \quad \forall j \in J, \forall k \in P_j \tag{5}$$

The objective function (1) seeks to minimize the weighted sum of the facility costs and the routing costs. Constraints (2) are the set partitioning constraints that require each customer i be served by exactly one of the selected routes. Constraints (3) require that facility j be selected if a route k associated with facility j is selected. Constraints (4) and (5) are standard binary restrictions.

The LRP with distance constraints is NP-hard. By placing very large costs on the arcs connecting two customer nodes, we obtain a special case of the model in which the selected routes contain exactly one customer. This special case is equivalent to the uncapacitated fixed charge facility location problem, which is well-known to be NP-hard (Cornuejols et al., 1990).

As presented, the formulation LRP-DC potentially contains an exponential number of variables (Y_{jk}) and an exponential number of constraints (3). Thus, for instances of practical size, enumerating all of the feasible routes and solving the resulting integer program is unlikely to be effective. Instead, we will use column generation in combination with branch-and-bound to solve instances of our formulation. Column generation has become a popular technique for solving very large linear programs; the basic idea is to solve a linear program by generating a subset of the variables while guaranteeing optimality of the solution over the set of all variables. Branch-and-bound is a standard technique for solving integer programs in which a series of relaxations of the original problem is solved to obtain an optimal solution. Branch-and-price is the name given to the generalized version of branch-and-bound in which the LP relaxation at each node of the tree is solved via column generation. We describe the details of our branch-and-price algorithm in §2.

1.2 Valid Inequalities

One key to an effective branch-and-bound algorithm is a strong lower bound. Unfortunately, the LP relaxation of LRP-DC provides a weak lower bound. We typically observe LP relaxation solutions in which the route variables are assigned very small fractional values and the facility variables are assigned fractional values large enough only to satisfy the $X_j - Y_{jk} \geq 0$ constraints. As a result, the objective function value of the LP relaxation may be a small percentage of the optimal integer objective function value.

To preclude such fractional solutions, we consider the following set of valid inequalities for the integer program:

$$X_j - \sum_{k \in P_j} a_{ijk} Y_{jk} \geq 0 \quad \forall i \in I, \forall j \in J \tag{6}$$

In a feasible integer solution, each customer $i \in I$ is on exactly one path, so $\sum_{k \in P_j} a_{ijk} Y_{jk}$ equals 1 for some facility j and 0 for all others. For any facility j such that $\sum_{k \in P_j} a_{ijk} Y_{jk}$ equals 1, X_j will be 1, which corresponds to routes being selected only for selected facilities. Berger (1997) shows that these new constraints are rank-1 Chvátal cuts.

The new set of constraints (6) implies the previous set (3) independent of the integrality constraints, because $Y_{jk} \geq 0$ and $a_{ijk} \geq 0$. For each facility $j \in J$ and route $\hat{k} \in P_j$, let i be some customer on route $\hat{k}$. Then,

$$X_j \geq \sum_{k \in P_j} a_{ijk} Y_{jk} \geq a_{ij\hat{k}} Y_{j\hat{k}}$$

Therefore, we can replace constraints (3) by constraints (6) in the formulation LRP-DC. Note that the resulting formulation has a polynomial number of constraints, since we removed $\sum_{j \in J} |P_j|$ constraints and added $|J| \cdot |I|$ constraints. In addition, the LP relaxation bound of the model with constraints (6) is greater than or equal to that of original model, because the feasible region of the LP relaxation of the model with constraints (6) is contained in the feasible region of the original model and the objective coefficients are the same. Throughout the remainder of the paper, all references to LRP-DC refer to the formulation with constraints (6) instead of constraints (3). In §3, we will illustrate empirically the improvement in the LP relaxation bound.

2 Branch-and-Price Algorithm

In this section, we describe a branch-and-price algorithm for the exact solution of the LRP with distance constraints. In recent years, branch-and-price has been applied successfully to solve problems in urban transit crew scheduling (Desrochers and Soumis, 1989), airline crew scheduling (Vance, 1993), vehicle routing (Desrosiers et al. 1984, Desrochers et al. 1992), pickup and delivery with time windows (Dumas et al. 1991, Savelsbergh and Sol 1998) and generalized assignment (Savelsbergh, 1997). Barnhart et al. (1998) present a discussion of the general methodology of branch-and-price.

2.1 Column Generation

Column generation has become a widely used technique for solving large-scale linear programs. The basic concept of column generation is to solve a linear program by generating only a subset of the variables while guaranteeing optimality of the solution over the entire set of variables.

For the LRP-DC model, we use column generation to solve the LP relaxation. We use the term *restricted master problem* (RMP) to refer to a restricted version of the LP relaxation that contains only a subset of the feasible route variables (but all of the location variables). We solve an optimization problem called the *pricing problem* to identify columns to add to the RMP. From linear programming theory, we know that the reduced cost of every variable in an optimal solution is non-negative (for minimization). Therefore, if the pricing problem identifies feasible route columns with negative reduced cost, we add them to the RMP and reoptimize. We stop the column generation procedure when the pricing problem identifies no columns to add; that is, every feasible route variable has a non-negative reduced cost, which implies that the optimal solution to the RMP is optimal for the original problem.

2.1.1 Formulating the Pricing Problem

The objective of the pricing problem for LRP-DC is to determine if there exists a feasible route k associated with some candidate facility j that has a negative reduced cost. To compute the reduced cost, we associate the dual variable π_i with the partitioning constraint (2) for customer i and the dual variable μ_{ij} with the constraint (6) corresponding to customer i and facility j. Then, from LP duality theory, we know that the reduced cost of a route k associated with facility j, denoted $\hat{c}_{jk}$,

$$\hat{c}_{jk} = c_{jk} - \sum_{i \in I} a_{ijk}(\pi_i - \mu_{ij})$$

Recall that the cost c_{jk} of a route is the sum of the costs of the arcs in the route. That means that if we can transfer the dual variable information currently associated with the nodes to the arcs, then we can compute the reduced cost of a route as the sum of the reduced costs of the arcs in the route. Then, we can identify eligible columns to enter the basis by solving a distance-constrained shortest path problem for each facility $j \in J$ over a network with the arc costs equal to the reduced costs of the arcs.

To construct the subproblem network for each facility $j \in J$, we define a subgraph $G^j \subset G$. Let $N^j = \{0, n+1\} \cup I$ denote the set of nodes. Node 0 represents facility j as the starting point and node $n+1$ represents facility j as the ending point of a route. (We assume that $|I| = n$.) Let $A^j = (\{0\} \times I) \cup (I \times I) \cup (I \times \{n+1\})$ be the set of arcs. Let q_{lm} denote the cost of arc $(l, m) \in A$. Then, the cost of arc $(l, m) \in A^j$ is the reduced cost of the arc:

$$\hat{q}_{lm} = \begin{cases} q_{lm} & \text{if } m = n+1 \\ q_{lm} - \pi_m + \mu_{mj} & \text{otherwise} \end{cases}$$

In this way, we have transferred the dual variable information from the nodes to the arcs. Then, to determine whether there are negative reduced cost route columns not in the current RMP, we solve a distance-constrained shortest path problem for each facility $j \in J$. That is, for each facility $j \in J$, we solve a shortest path problem over the subgraph $G^j = (N^j, A^j)$ with arc costs $\hat{q}_{lm}$ for all

$(l, m) \in A^j$ and with the restriction that the length of the route be at most M. Let z_j^* denote the (reduced) cost of the solution of the pricing problem for facility j. If for all $j \in J$, $z_j^* \geq 0$, the LP relaxation (the master problem) has been solved. Otherwise, for each $j \in J$ with $z_j^* < 0$, we add the corresponding column to the RMP and reoptimize.

2.1.2 Solving the Pricing Problem

The solution of the pricing problem for facility j specifies a feasible path of minimum cost with respect to the *reduced cost* on the arcs. More specifically, we seek the solution of an elementary shortest path problem with a single resource constraint (total distance) over a graph G^j that may contain negative cost cycles. One approach to solving the pricing problem is to solve a relaxed version of the problem in which we allow solutions to possibly visit customer nodes more than once. Columns of this type can be represented in the RMP by letting a_{ijk} denote the number of times a node i is visited by route k associated with facility j. This relaxed version of the pricing problem can be solved efficiently using a standard forward dynamic programming algorithm (e.g. Ahuja et al., 1993). A significant disadvantage of the approach is that the lower bound is weakened, as reported for the VRPTW and for other vehicle routing problems (Feillet et al. 2004, Gelinas et al. 1995). As in the context of VRPTW, we could modify the algorithm to eliminate cycles (Desrochers et al. 1992, Irnich and Villeneuve 2005) or add valid inequalities (Kohl et al., 1999). Instead, however, we use Feillet et al. (2004)'s extended label correcting algorithm to solve an elementary shortest path problem with resource constraints for each facility $j \in J$.

In describing how to apply Feillet et al.'s algorithm to the pricing problem, we follow their notation and terminology. We associate a label λ_i with each path X_{pi} from the source node to a node v_i. Label λ_i has a cost C_i, which is the reduced cost of the path, and a state $R_i = (T_i, s_i, V_i^1, V_i^2, \cdots, V_i^n)$, where T_i denotes the length of the path (quantity of the resource consumed), s_i denotes the number of "unreachable" nodes and $V_i^1, V_i^2, \cdots, V_i^n$ is the vector of "unreachable" nodes. In the path X_{pi} from p to v_i, a node v_k is said to be *unreachable* if it is included in the path or if extending the path from v_i to v_k would violate the distance constraint ($T_i + d_{ik} > M$). In the vector of unreachable nodes, V_i^k is assigned a value of one if node v_k is unreachable from v_i and a value of zero otherwise. We let Λ_i denote the set of all labels associated with node v_i.

The basic idea of the algorithm is to repeatedly examine the nodes of the graph, each time extending each of the partial paths (labels) toward all possible successor nodes until no new labels are created. When examining a node v_i, we consider the list of successor nodes $\text{succ}(v_i)$. For each $v_j \in \text{succ}(v_i)$, we create a set of labels F_{ij}, which contains the labels extended from node v_i to node v_j. If a label $\lambda_i \in \Lambda_i$ can be extended from v_i to v_j, a new label is created and added to F_{ij}. A label can be extended from v_i to v_j if v_j is not unreachable ($V_i^j = 0$) and $T_i + d_{iv_j} \leq M$; the values stored in the label correspond to the reduced cost of extending the path to v_j, the updated length of the route, the updated number of unreachable nodes and the updated vector of unreachable nodes. After considering the extension of all of the labels to all of the possible successor nodes, we merge the new set of labels F_{ij} with the set of existing labels on node v_j, Λ_j. In merging the two sets, we apply domination rules to limit the growth in the number of labels. Consider two labels on node v_j. Let $\lambda_j = (C_j, R_j)$ with $R_j = (T_j, s_j, V_j^1, V_j^2, \cdots, V_j^n)$ and let $\hat{\lambda}_j = (\hat{C}_j, \hat{R}_j)$ with $\hat{R}_j = (\hat{T}_j, \hat{s}_j, \hat{V}_j^1, \hat{V}_j^2, \cdots, \hat{V}_j^n)$. Label λ_j dominates $\hat{\lambda}_j$ if $C_j \leq \hat{C}_j$, $T_j \leq \hat{T}_j$, $s_j \leq \hat{s}_j$ and $V_j^k \leq \hat{V}_j^k$ for all $k = 1, \cdots, n$ and $(C_j, R_j) \neq (\hat{C}_j, \hat{R}_j)$. The domination rule says that a label λ_j dominates a second label $\hat{\lambda}_j$ if the first one is not more expensive, is not longer, has no more unreachable nodes and does not have an unreachable node not shared by the second label; the two labels must be different in at least one value. In the final step of an iteration, if the set of labels Λ_j changed during the course of examining node v_i, we add node v_j to the list of nodes to be examined.

The algorithm terminates when no new labels can be created for any of the nodes. Then, we examine the set of labels associated with the sink node Λ_{n+1}. Each $\lambda \in \Lambda_{n+1}$ corresponds to a feasible route that is not dominated by any other route; any non-dominated route with negative reduced cost is a candidate column to add to the RMP.

2.2 Branching Rules

An optimal solution of the LP relaxation may contain variables with fractional values. Applying a standard branch-and-bound procedure does not guarantee optimality since the column set does not contain all of the variables. After variables are fixed during the branching process and the LP is reoptimized, the dual variables may have new values. As a result, a column that did not price out favorably at a previous node of the tree may now have negative reduced cost. Therefore, additional columns must be generated throughout the tree.

Branch-and-price is the name given to the generalized version of branch-and-bound in which column generation is used to solve an appropriate linear program at each node of the tree. In developing a branching strategy for a branch-and-price algorithm, we need to consider the effects of the branching rule on the solution of the pricing problem. For LRP-DC, our objective is to ensure that we can continue to solve the modified pricing problem that includes the branching decisions as a shortest path problem. We develop different branching rules for the two variable classes.

For the location variables X_j, conventional variable dichotomy branching is effective. The constraint $X_j = 1$ is easily incorporated into the RMP by fixing X_j to 1 and then solving the pricing problem for facility j as before. The constraint $X_j = 0$ can be achieved by fixing X_j to 0 and Y_{jk} to 0 for all corresponding routes $k \in P_j$ in the RMP and then not solving the pricing problem for facility j.

For the route variables Y_{jk}, we use the branching rule suggested by Desrochers and Soumis (1989) that is a modification of the Ryan and Foster (1981) rule. The main idea of the modified rule is to branch on a pair of customers that appear *consecutively* in a path instead of on a pair of customers that appear in the same path in any order. As noted by Desrochers and Soumis (1989), the modified rule is easier to implement within a column generation algorithm. To apply the modified rule to LRP-DC, we define $T_j(t_1, t_2)$ to be the set of routes associated with facility j in which customer t_2 immediately follows customer t_1. We choose a pair of customers t_1 and t_2 such that in the current LP solution, $0 < \sum_{j\in J} \sum_{k\in T_j(t_1,t_2)} Y_{jk} < 1$. Then, we create two branches:

$$\begin{array}{lrcl} \text{0-branch} & \sum_{j\in J} \sum_{k\in T_j(t_1,t_2)} Y_{jk} & = & 0 \\ \text{1-branch} & \sum_{j\in J} \sum_{k\in T_j(t_1,t_2)} Y_{jk} & = & 1 \end{array}$$

In the 0-branch, we prohibit any route from visiting consecutively customers t_1 and t_2. In the RMP, we fix to 0 any columns that visit customers t_1 and t_2 consecutively, i.e., $Y_{jk} = 0$ $\forall k \in T_j(t_1, t_2)$ and $\forall j \in J$; in the pricing problem, we delete the arc between customers t_1 and t_2 to prohibit generating new columns that violate the branching decision. In the 1-branch, we require any route visiting customer t_1 to visit t_2 immediately. In the RMP, we fix to 0 any columns that cover either t_1 or t_2 alone or that cover both t_1 and t_2 in a different order; in the pricing problem, we combine customers t_1 and t_2 into a single node to facilitate generating new columns that satisfy the branching decision. Note that our branching rules are equivalent to forbidding or requiring an arc in a route in the context of a vehicle flow formulation for the LRP with distance constraints.

2.3 Implementation Details

We implement our branch-and-price algorithm in MINTO 3.1 (Savelsbergh and Nemhauser, 1996) with CPLEX 8.1. MINTO is a general purpose mixed integer optimizer that can be customized by incorporating application-specific functions. For our branch-and-price algorithm, we develop specific routines for creating an initial solution, solving the pricing problem, generating an upper bound at the root node and choosing a branching variable. We provide some details of each function in this section.

Initial Feasible Solution To begin the algorithm, we need an initial set of columns that contains a feasible solution. One possibility is to construct the set of all feasible *singleton* routes – routes that visit a single customer with a roundtrip distance of at most M. The disadvantage of this simple approach is that the initial solution is likely to be far from optimal, and, as a result, relatively more iterations of pricing may be required to solve the LP relaxation. To improve the quality of the initial starting solution, we construct a set of feasible routes that involve multiple customers in addition to the singleton routes. To construct these routes, we use a nearest insertion heuristic procedure.

For each facility j, we start with a route that includes only customer node $i = 1$ and repeatedly add the customer node of minimum insertion distance until we reach the maximum allowable number of customers, which is a parameter in the algorithm, or we reach the maximum distance M. Then we repeat the procedure giving each customer node i for $i = 2, \cdots, |I|$ the opportunity to be the "seed" node of the route. We store the routes in order of increasing average cost, defined as the total cost of the route divided by the number of customers served. At the end of the procedure, we calculate a threshold value and add any column with an average cost less than the threshold to the initial set of columns. In our computational experiments, we construct routes with up to four customers, and we compute the threshold value as the midpoint between the minimum average cost and the maximum average cost of the routes constructed.

Pricing Problem Recall that the pricing problem decomposes into a set of independent pricing problems, one for each facility j. That means that during each pricing iteration, we must solve $|J|$ elementary shortest path problems with one resource constraint. The label correcting algorithm for solving the ESPPRC typically yields multiple non-dominated routes associated with a facility; we add all of the non-dominated routes with negative reduced cost to the RMP. During an iteration of the column generation algorithm, there may be a facility j for which no negative reduced cost columns are generated. If the dual variable values are not changing much from iteration to iteration (because the solution is close to optimal), then it is likely that no negative reduced cost columns will be generated for facility j in the next iteration. Therefore, in hopes of reducing the time spent generating columns, we do not solve the pricing problem for facility j in the next iteration. To prove optimality, however, the pricing problem for all facilities must be solved during the final iteration.

Primal Feasible Solution Before beginning the branching portion of the branch-and-price algorithm, it is important to have good lower and upper bounds on the optimal integer objective function value. The objective function value of the optimal LP relaxation solution is a valid lower bound. To obtain an upper bound at the root node, we apply a standard branch-and-bound algorithm to solve the integer program defined over the set of columns generated during the solution of the LP relaxation. In general, this upper bound is excellent, helping to keep the tree size manageable. At subsequent nodes of the tree, the algorithm calls the default MINTO primal heuristic that tries to find a better integer solution.

Branch Selection Given a fractional solution at a node of the branch-and-bound tree, we first branch on a fractional location variable X_j. Given more than one fractional location variable, we branch on the most fractional variable, an X_j with the value closest to 0.5. In general, forcing the location variables to be integer significantly reduces the number of fractional route variables. Next, we branch on an ordered pair of customers t_1 and t_2 as described in §2.2, selecting the pair such that the value of $\sum_{j \in J} \sum_{k \in T_j(t_1,t_2)} Y_{jk}$ is closest to 0.5.

3 Computational Results

In this section, we present the results of our computational experiments with the LRP-DC formulation presented in §1 and the branch-and-price algorithm described in §2. We ran all of the computational experiments on a Linux-based workstation with a 2.4 GHz processor and 2 GB RAM. We terminated the branch-and-price algorithm when either we found a provably optimal integer solution or we exceeded 2 hours of CPU time.

3.1 Data

To specify an instance of the LRP with distance constraints, we need the locations of the customers and the candidate facilities, the associated distance matrix, the facility costs, the routing costs, a value for the objective weighting factor α and a value for the maximum distance M. We describe the details of each element in turn.

Location Data For our study, we created a set of 8 base instances, each with 100 customers and 10 candidate facility locations. The customer locations are generated from the Solomon VRPTW instances (Solomon, 1987). The Solomon instances are divided into six groups, denoted R1, R2, C1, C2, RC1 and RC2, and for all of the instances within a group, the customer locations are the same. In R1 and R2, the customer locations are randomly generated from a uniform distribution, and in C1 and C2, they are clustered. In RC1 and RC2, the customer locations are a combination of randomly generated and clustered points. Because the (x, y) coordinates of the customer locations are the same for R1 and R2 and for RC1 and RC2, the Solomon instances yield only four sets of distinct customer locations. For the LRP with distance constraints, we created one instance corresponding to each group of customer locations, R1, C1, C2 and RC1 respectively.

For the candidate facility locations, we created two sets of 10 candidate locations – the same two sets are used for all four of the customer instances. We randomly generated the candidate locations from a uniform distribution, and we continued generating candidate locations until we obtained a feasible set; that is, each customer location could be reached by a singleton route of at most $M = 40$ units from at least one candidate facility.

By pairing each set of customer locations with each set of candidate locations, we obtained 8 base instances. Associated with each of these 8 instances are four additional instances (for a total of 40 instances) generated by considering subsets of the customer locations – the first 50, the second 50, the first 75 and the second 75 respectively. We identify each instance by an ID. The first part of the ID specifies the problem group, R1, C1, C2 or RC1; the second part specifies the customer subset, the first 50 (50a), the second 50 (50b), the first 75 (75a), the second 75 (75b) or all 100 (100); and the third part specifies the set of candidate facility locations, 1 or 2.

Distance Matrix To calculate the distance matrix, we begin by calculating the Euclidean distance between all node pairs and rounding the value to the nearest integer. Next, we use the Floyd-Warshall algorithm (Ahuja et al., 1993) to calculate the shortest path distances for all node pairs. We assign the distance between a pair of nodes to be the corresponding shortest path distance, and we allow links between any node pair for which the distance is less than or equal to the maximum allowable distance M.

Costs We expect the strength of the formulation and the effectiveness of the branch-and-price algorithm to be influenced by the *relative* costs of the facilities and the routes. We use the objective weighting factor α to change the weight of the facility costs. Facility costs vary widely depending on the type of facility, the location of the facility and the size of the facility. For the base case ($\alpha = 1$), we randomly generated facility costs in the range of \$55 to \$110, which corresponds to a daily equivalent cost of an annual lease on a 10,000 sq. ft. facility at \$200 to \$400 per sq. ft. (Burlett, 2002). For routing costs, we assume a cost of \$1 per unit distance to cover variable vehicle costs; we assume that personnel costs and other vehicle related fixed costs are considered outside the location-routing decision. To investigate the impact of relatively more expensive facility costs, we also consider an objective weighting parameter value of $\alpha = 5$.

Maximum Distance Recall that we motivated our research in the introduction with examples of LRPs that might arise in the operations of organizations such as Meals on Wheels, Barnesandnoble.com and FedEx. Other applications that have distance constraints include school bus routes, blood donation collection and oil and gas delivery. Since each application tends to have a different number of stops per route and a different maximum distance, we tested our algorithm for maximum distance values of $M = 40$ and $M = 60$ for all of the instances. For the 50 customer instances, we also tested the algorithm for $M = 80$, although for some instances, even the root node could not be solved within the CPU limit of two hours. The difficulty increases with the maximum distance since there are more feasible routes.

3.2 Results

Before we discuss the performance of the branch-and-price algorithm, we illustrate empirically the improvement in the LP relaxation bound achieved by the substitution of the valid inequalities (6) for the constraints (3). We use the terminology *weak relaxation* to refer to the LP relaxation of the formulation with constraints (3) and *strong relaxation* to refer to relaxation of the formulation with constraints (6). Table 1 shows the objective function values of the two relaxations as a ratio of the optimal integer objective function value, or the best integer solution obtained within 2 hours (indicated by a star). To obtain the weak LP relaxation bound, we enumerated all of the feasible routes with $M = 40$ for all of the instances and solved the LP relaxation using MINTO. Even for $M = 40$, some of the instances required too much memory to be solved; we indicate these instances in the table by a dash. The strong LP relaxation bounds were obtained using the branch-and-price algorithm.

Overall, the table shows that the strong relaxation is much better than the weak relaxation. Even taking into account the instances for which the optimal integer solution is not known, the strong LP bound ratios range from 0.928 to 1 and average above 0.99, and the ratio does not seem to be affected by the value of α. The weak LP bound ratios range from 0.439 to 0.995 for $\alpha = 1$ and from 0.137 to 0.998 for $\alpha = 5$; in general, the larger α value results in a lower ratio for the weak relaxation. As we will show in the next set of tables, the strong relaxation ratio remains high as the maximum distance increases.

In discussing the performance of the branch-and-price algorithm, we use the following notation throughout the remainder of this section.

LP/IP ratio of the strong LP relaxation bound to the optimal integer objective function value

Gap integrality gap after 2 hours of CPU time expressed as a percentage; the gap is zero if there is no value reported

Nodes number of branch-and-bound nodes evaluated

Cols total number of columns generated

CG CPU time (in seconds) spent solving the pricing subproblems

LP CPU time (in seconds) spent solving the master problems

Total total CPU time (in seconds)

#X number of locations selected in the optimal solution

	DC Set 1				DC Set 2			
	$\alpha = 1$		$\alpha = 5$		$\alpha = 1$		$\alpha = 5$	
ID	Weak	Strong	Weak	Strong	Weak	Strong	Weak	Strong
R1-50a	0.956	0.999	0.436	1	0.605	1	0.284	1
R1-50b	0.560	1	0.286	1	-	1	-	1
R1-75a	0.959	0.995	0.256	1	0.633	0.998	0.247	0.998
R1-75b	-	0.998	-	1	-	1	-	0.999
R1-100	-	0.989*	-	0.997	-	0.993	-	0.998
C1-50a	0.461	0.928*	0.180	0.973*	-	0.981	-	0.994
C1-50b	0.549	0.998	0.268	0.999	0.502	0.993	0.2528	0.997
C1-75a	0.561	0.972*	0.292	0.989*	-	0.991	-	0.995
C1-75b	0.558	0.988	0.269	0.996	-	0.993	-	0.997
C1-100	0.517	0.973*	0.239	0.992*	-	0.989	-	0.996
C2-50a	0.540	0.998	0.199	0.999	0.549	0.997	0.232	0.998
C2-50b	0.995	0.995	0.998	0.998	0.547	0.989	0.279	0.991
C2-75a	0.611	0.998	0.293	0.999	0.574	0.995	0.258	0.998
C2-75b	0.670	0.996	0.396	0.998	0.612	0.998	0.287	0.999
C2-100	0.623	0.997	0.291	0.999	0.590	0.998	0.266	0.999
RC1-50a	0.439	0.993	0.137	0.998	0.522	0.993	0.196	0.998
RC1-50b	0.659	0.999	0.367	1	0.583	0.994	0.355	1
RC1-75a	0.576	1	0.319	0.999	0.641	0.994	0.390	0.997
RC1-75b	0.654	1	0.370	1	0.607	0.999	0.340	0.999
RC1-100	0.591	0.997	0.257	0.999	0.628	0.997	0.340	0.998
Avg	0.638	0.991	0.326	0.997	0.584	0.995	0.287	0.998

Table 1: Comparison of LP Relaxation Bounds

#Y number of routes selected in the optimal solution

Avg average number of customers on a route

In the tables, a value of zero for any of the CPU times indicates a time less than 0.5 seconds. For the instances in which there is an integrality gap at termination, the LP/IP ratio, the #X and the #Y are reported in reference to the best integer feasible solution.

R1 Instances Table 2 shows the results for the randomly generated instances. Some of the R1 instances were difficult for the branch-and-price algorithm. In particular, not even the root node of the 100 customer instances with $M = 60$ could be solved within two hours of CPU time, nor could three of the 50b instances with $M = 80$; to save space, we omit the 50b instances from the table. Otherwise, all but seven instances were solved to optimality within the CPU limit. Of the seven, six had integrality gaps of at most 0.4% and the other one had a gap of 1.20%. For 14 instances, the LP relaxation solution was optimal, and the LP relaxation bound remains strong as the maximum distance increases. For all but a few instances, the LP/IP ratio is above 0.99. As expected, the solution time, the number of branch-and-bound nodes and the number of columns generated increase with the maximum distance M. In terms of the solutions, the number of selected facilities decreases and the average number of customers per route increases as M increases. Both values, however, tend to remain relatively steady (for a given M) as the number of customers increases; this observation makes sense given that locations of the 50 customers span the same 100×100 area as the 100 customers. Increasing the relative weight of the facility costs changes both the number of selected facilities and the number of routes. When $\alpha = 5$, the number of selected facilities tends either to drop by one or remain unchanged but with at least one different facility selected. Changing the set of locations impacts the set of feasible routes, which results in the potential increase or decrease in the number of routes.

C1 Instances Table 3 shows the results for the C1 clustered instances. The customer locations in these instances are clustered fairly tightly in groups ranging in size from eight to 12 customers. Because of the tight clustering, the number of customers on a route can be large even when $M = 60$. As a result, the root node of the 50a instances with $M = 80$ could not be solved within two hours of CPU time; to save space, we omit the 50a instances from the table. Otherwise, all but 12 instances were solved to optimality within the CPU limit, however, the integrality gaps were much larger than those for the R1 instances. Four of the instances had gaps at most 0.67% but three of the instances had gaps larger than 4%. For nine of the instances, the LP relaxation solution was optimal. Overall, the LP relaxation bound appears to remain strong – the LP/IP ratio averages

0.992 excluding the instances with integrality gaps – although it appears to decrease slightly as M increases. For the second set of candidate facility locations, increasing the value of M results in longer CPU times and many more generated columns but not necessarily more branch-and-bound nodes. For the first set of candidate facility locations, the relationship is not as clear; increasing the value of M tends to result in more columns generated but not necessarily longer CPU times or more evaluated nodes. In terms of the solutions, the number of selected facilities decreases and the average number of customers per route increases as M increases. Unlike the R1 instances, however, for a given M, the number of facilities increases as the number of customers increases. (This occurs because the number of clusters increases as the number of customers increases and hence the area spanned by the customer locations increases.) Notice that the average number of customers per route is much larger for these instances than for the R1 instances. Increasing the relative weight of the facility costs did not result in many changes to the set of selected facilities. For the first set of candidate locations, the only changes occurred for the 100 customer instances; the number of locations remained the same but the locations selected differed by two for $M = 40$ and by one for $M = 80$. For the second set of candidate locations, the number of locations decreased by one for 50a with $M = 60$ and 50b with $M = 80$, and the selected locations differed by one for 50b with $M = 60$ and 75a with $M = 40$.

C2 Instances Table 4 shows the results for the C2 clustered instances. The customer locations in these instances are clustered more loosely than in the C1 instances. Overall, these instances, including all of the 50 customer instances with $M = 80$, were handled well by the branch-and-price algorithm. All but six instances were solved to optimality within two hours of CPU time. Of the six, four had integrality gaps of at most 0.55% and the others had gaps of 1.46% and 1.85%. For 10 of the instances, the LP relaxation solution is optimal and the LP/IP ratio remains strong. Both the solution times and the number of columns generated increase significantly as M increases but there is no clear relationship between the number of branch-and-bound nodes and M. As with the other problem classes, the number of selected facilities decreases and the average number of customers per route increases as M increases. Similar to C1, for a given M, the number of facilities increases as the number of customers increases. Not surprisingly, the average number of customers per route is between the values for the R1 and C1 instances. Increasing the relative weight of the facility costs had some effect on the number and the set of selected locations for the smaller instances. For both sets of candidate locations, almost all of the 50a and 50b instances had one fewer selected facilities or one different selected facility. For instance C2-75a-1, the larger value of α resulted in both one fewer and one different facility.

RC1 Instances Table 5 shows the results for the RC1 instances. The 100 customer locations in these instances are a combination of randomly generated and clustered points. For the 50 customer instances, 50a consists of five clusters while 50b consists of randomly generated points. For the 75 customer instances, 75a consists of five clusters plus 25 randomly generated points while 75b consists of randomly generated points plus 2 clusters. Overall, the branch-and-price algorithm was able to solve all of these instances to optimality easily; the longest CPU time was 640 seconds. The LP relaxation solution was optimal for 19 instances and the LP/IP ratio averaged 0.995 for the remaining instances. The solution times and the number of columns generated increases as M increases. For the 50a and 75a instances, the values for the number of selected facilities and the average number of customers per route are similar to the values for the clustered instances, whereas for the 50b and the 75b instances, the values are similar to those for the random instances. For the 100 customer instances, the number of selected facilities tends to be larger and the average number of customers per route tends to be smaller than for the other problem types. Increasing the relative weight of the facility costs resulted in a number of changes to the number and the set of selected facilities, particularly for the larger values of M. Instances RC1-50b-1 with $M = 80$ and RC1-100-1 with $M = 60$ had both one fewer and one different selected locations, while RC1-50a-1 with $M = 80$ had two fewer and two different. Instances RC1-50a-2 and RC1-100-2 with $M = 60$ have one different selected location and RC1-50b-2 with $M = 60$ has two different selected locations.

4 Summary

In this paper, we presented a new set-partitioning-based formulation for a class of location-routing problems with distance constraints. We presented a set of valid inequalities that strengthens the formulation and demonstrated the improvement empirically for a set of small instances. We developed a branch-and-price algorithm that uses an extended label correcting algorithm for the ESPPRC (Feillet et al., 2004) to solve the pricing problem and presented results for the algorithm. Due in part to the strength of the LP relaxation bound, the algorithm was able to solve optimally instances with as many as 100 customer nodes and 10 candidate facilities with various distance constraints.

The structure of the set-partitioning-based formulation is such that additional or different constraints on the vehicle routes can be added easily. For example, adding vehicle capacity constraints adds another dimension to the definition of a feasible route associated with a candidate facility. The pricing subproblem for each facility becomes an elementary shortest path problem with two resource constraints that can still be solved by the extended label correcting algorithm of Feillet et al. (2004). The difficulty of solving instances with additional or different resource constraints depends in part on whether, and how much, the additional restrictions reduce the solution space and whether the LP relaxation bound remains strong. Future work also will investigate adding cutting planes to the branch-and-price algorithm to allow even larger instances to be solved effectively.

Acknowledgments This research was supported in part by grants DDM-9224998 and DMI-9523048 from the National Science Foundation.

ID	MAX	α	LP/IP	Gap	Nodes	Cols	CG	LP	Total	# X	# Y	Avg
R1-50a-1	40	1	1		1	204	0	0	0	8	20	2.50
	60	1	1		1	714	0	0	1	5	12	4.17
	80	1	0.984		61	2710	100	3	153	3	8	6.25
R1-50b-1	40	1	1		1	525	5	0	5	7	16	3.13
	60	1	0.979	1.20	47	3529	7030	6	7200	5	12	4.17
R1-75a-1	40	1	0.996		27	608	1	0	2	9	29	2.59
	60	1	0.990		193	3862	525	18	1976	5	13	5.77
R1-75b-1	40	1	0.999		5	682	56	0	56	9	30	2.50
	60	1	0.994	0.11	19	2587	7190	1	7200	5	14	5.36
R1-100-1	40	1	0.990	0.40	1020	2761	1916	13	7200	8	38	2.63
	60	1										
R1-50a-1	40	5	1		1	205	0	0	0	7	23	2.17
	60	5	0.998		51	750	2	0	4	4	14	3.57
	80	5	0.986	0.16	514	3373	234	20	7200	2	10	5.00
R1-50b-1	40	5	1		1	532	5	0	5	6	18	2.78
	60	5	0.980		321	5715	1522	13	7200	4	18	2.78
R1-75a-1	40	5	1		1	441	0	0	0	8	30	2.50
	60	5	0.996		39	2025	65	1	94	5	13	5.77
R1-75b-1	40	5	1		3	836	31	0	31	8	26	2.88
	60	5	0.998		23	3157	7197	2	7200	5	18	4.17
R1-100-1	40	5	0.997		489	2506	1355	7	1935	8	32	3.13
	60	5										
R1-50a-2	40	1	1		1	235	0	0	0	7	19	2.63
	60	1	1		1	824	0	0	1	4	11	4.55
	80	1	0.998		3	2089	18	1	19	3	8	6.25
R1-50b-2	40	1	1		2	482	3	0	3	6	17	2.94
	60	1	0.995		29	2280	358	1	375	4	11	4.55
R1-75a-2	40	1	0.998		24	577	2	0	4	8	25	3.00
	60	1	0.990	0.30	370	4483	384	29	7200	4	14	5.36
R1-75b-2	40	1	1		7	794	21	0	22	8	24	3.13
	60	1	0.987	0.25	84	3324	6931	5	7200	4	14	5.36
R1-100-2	40	1	0.993		5	1227	122	1	123	8	28	3.57
	60	1										
R1-50a-2	40	5	1		1	238	0	0	0	7	21	2.38
	60	5	1		1	804	0	0	0	4	12	4.17
	80	5	0.998		9	2749	30	1	35	2	8	6.25
R1-50b-2	40	5	1		1	478	3	0	3	6	17	2.94
	60	5	0.999		35	2120	266	1	305	4	10	5.00
R1-75a-2	40	5	0.998		15	553	1	0	1	7	29	2.59
	60	5	0.995	0.12	616	4265	246	25	7200	4	15	5.00
R1-75b-2	40	5	0.999		7	817	8	0	8	7	30	2.50
	60	5	0.996		177	4096	2749	8	3718	4	16	4.69
R1-100-2	40	5	0.998		435	2109	218	5	581	7	30	3.33
	60	5										

Table 2: Branch-and-Price Results for R1 Problem Instances

ID	MAX	α	LP/IP	Gap	Nodes	Cols	CG	LP	Total	# X	# Y	Avg
C1-50a-1	40	1	0.928	4.48	864	3435	225	10	7200	3	10	5.00
	60	1	1		1	1735	81	0	81	2	5	10.00
C1-50b-1	40	1	0.999		5	449	0	0	1	6	14	3.57
	60	1	0.992		5	1180	9	0	10	3	12	4.17
	80	1	1		3	2610	516	0	517	3	5	10.00
C1-75a-1	40	1	0.972	1.43	711	3167	669	9	7200	6	16	4.69
	60	1	1		1	2582	595	0	596	4	9	8.33
C1-75b-1	40	1	0.988		349	2128	45	4	344	8	21	3.57
	60	1	0.994		9	2300	456	1	457	4	13	5.77
C1-100-1	40	1	0.974	1.79	658	3128	578	10	7200	8	20	5.00
	60	1	0.982		35	3629	4228	2	4241	5	12	8.33
C1-50a-1	40	5	0.973	1.73	659	2121	624	10	7200	3	10	5.00
	60	5	1		1	1677	110	0	110	2	5	10.00
C1-50b-1	40	5	0.999		5	451	0	0	0	6	14	3.57
	60	5	0.997		3	1104	3	0	3	3	12	4.17
	80	5	1		2	2372	224	0	225	3	5	10.00
C1-75a-1	40	5	0.989	0.55	789	1585	756	11	7200	6	17	4.41
	60	5	1		1	2639	791	0	792	4	9	8.33
C1-75b-1	40	5	0.996		269	2040	38	3	260	7	21	3.57
	60	5	0.997		7	2177	136	0	137	4	13	5.77
C1-100-1	40	5	0.992	0.38	624	2593	673	12	7200	8	20	5.00
	60	5	0.988	0.67	196	5333	5458	10	7200	5	22	4.55
C1-50a-2	40	1	0.981		42	1228	16	1	25	4	8	6.25
	60	1	0.893	5.69	251	5100	3314	18	7200	3	6	8.33
C1-50b-2	40	1	0.993		17	545	1	0	1	5	13	3.85
	60	1	1		1	1015	2	0	2	4	7	7.14
	80	1	1		1	2534	2048	0	2049	3	5	10.00
C1-75a-2	40	1	0.992		51	1637	121	1	137	6	17	4.41
	60	1	0.963	2.35	21	4719	7122	4	7200	4	10	7.50
C1-75b-2	40	1	0.994		19	929	35	0	37	6	23	3.26
	60	1	0.975		64	5349	6177	7	6491	4	9	8.33
C1-100-2	40	1	0.989		229	2656	233	5	616	7	19	5.26
	60	1	0.971	2.51	7	4261	7197	2	7200	5	12	8.33
C1-50a-2	40	5	0.994		13	970	15	0	16	4	8	6.25
	60	5	0.924	5.90	217	4982	2941	11	7200	2	8	6.25
C1-50b-2	40	5	0.998		15	512	0	0	1	5	13	3.85
	60	5	0.940		21	1552	20	0	24	4	9	5.56
	80	5	1		1	2672	629	0	630	2	6	8.33
C1-75a-2	40	5	0.996		191	2191	104	3	211	6	15	5.00
	60	5	0.989		125	9039	4422	20	7008	4	10	7.50
C1-75b-2	40	5	0.998		27	934	43	0	44	6	17	4.41
	60	5	0.990		51	5696	1302	6	1452	4	9	8.33
C1-100-2	40	5	0.997		133	2371	253	4	416	7	24	4.17
	60	5	0.989	0.22	94	8395	5689	20	7200	5	12	8.33

Table 3: Branch-and-Price Results for C1 Problem Instances

ID	MAX	α	LP/IP	Gap	Nodes	Cols	CG	LP	Total	# X	# Y	Avg
C2-50a-1	40	1	0.999		7	537	1	0	1	6	11	4.55
	60	1	1		1	1274	21	0	21	4	8	6.25
	80	1	0.991		27	4272	3525	4	3579	2	6	8.33
C2-50b-1	40	1	0.996		7	323	0	0	0	6	19	2.63
	60	1	0.961		15	1108	6	0	7	4	9	5.56
	80	1	0.998		5	2760	664	1	666	2	7	7.14
C2-75a-1	40	1	0.998		7	696	2	0	2	8	18	4.17
	60	1	0.985		242	12619	1105	33	7117	5	12	6.25
C2-75b-1	40	1	0.997		9	625	1	0	1	8	20	3.75
	60	1	0.967	1.46	391	6679	616	20	7200	4	13	5.77
C2-100-1	40	1	0.997		17	914	3	0	3	9	30	3.33
	60	1	0.974	1.85	132	10234	2612	23	7200	5	16	6.25
C2-50a-1	40	5	0.999		5	500	0	0	1	6	15	3.33
	60	5	1		1	1330	10	0	11	3	9	5.56
	80	5	1		1	2874	551	1	553	2	6	8.33
C2-50b-1	40	5	0.999		7	325	0	0	0	6	19	2.63
	60	5	0.993		13	1081	2	0	2	3	13	3.85
	80	5	0.999		34	2686	873	1	914	3	6	8.33
C2-75a-1	40	5	0.999		7	696	2	0	2	8	18	4.17
	60	5	0.994	0.14	505	7485	779	23	7200	4	13	5.77
C2-75b-1	40	5	0.999		9	625	1	0	1	8	20	3.75
	60	5	0.988	0.38	492	4962	575	13	7200	4	13	5.77
C2-100-1	40	5	0.999		17	934	3	0	3	9	31	3.23
	60	5	0.990	0.55	322	5196	1143	11	7200	5	16	6.25
C2-50a-2	40	1	0.998		5	443	0	0	1	5	14	3.57
	60	1	0.999		1	1172	11	0	12	4	8	6.25
	80	1	1		1	2833	1535	1	1539	3	5	10.00
C2-50b-2	40	1	0.989		83	585	2	0	4	6	14	3.57
	60	1	0.979		55	1934	28	2	44	4	9	5.56
	80	1	0.996		11	3053	1319	2	1335	3	7	7.14
C2-75a-2	40	1	0.996		13	755	2	0	3	6	25	3.00
	60	1	0.999		1	2158	955	1	957	5	14	5.36
C2-75b-2	40	1	0.998		5	690	1	0	2	6	21	3.57
	60	1	1		1	2109	201	0	202	4	12	6.25
C2-100-2	40	1	0.998		5	956	1	0	2	7	30	3.33
	60	1	1		1	3039	405	1	407	5	14	7.14
C2-50a-2	40	5	0.998		15	442	0	0	1	5	15	3.33
	60	5	1		1	1160	9	0	9	4	8	6.25
	80	5	0.994		105	6397	2550	10	3409	2	7	7.14
C2-50b-2	40	5	0.992	0.04	2758	4744	23	26	7200	5	19	2.63
	60	5	0.949		3	1257	16	0	17	5	15	3.33
	80	5	0.997		5	3154	925	1	927	2	7	7.14
C2-75a-2	40	5	0.999		19	788	2	0	2	6	25	3.00
	60	5	1		1	2320	136	1	137	5	11	6.82
C2-75b-2	40	5	0.999		5	654	0	0	1	6	21	3.57
	60	5	1		1	2172	87	1	88	4	12	6.25
C2-100-2	40	5	0.999		5	958	2	0	2	7	30	3.33
	60	5	1		1	3063	185	1	186	5	14	7.14

Table 4: Branch-and-Price Results for C2 Problem Instances

ID	MAX	α	LP/IP	Nodes	Cols	CG	LP	Total	# X	# Y	Avg
RC1-50a-1	40	1	0.993	11	664	3	0	4	5	8	6.25
	60	1	1	1	930	3	0	3	5	5	10.00
	80	1	1	1	1380	4	0	4	5	5	10.00
RC1-50b-1	40	1	0.999	1	307	0	0	0	5	20	2.50
	60	1	0.995	13	1087	7	0	8	5	11	4.55
	80	1	1	1	2032	173	1	174	3	8	6.25
RC1-75a-1	40	1	1	1	766	1	0	1	8	23	3.26
	60	1	1	1	1330	3	0	4	6	13	5.77
RC1-75b-1	40	1	1	1	517	0	0	0	8	25	3.00
	60	1	0.999	5	1352	5	0	6	6	15	5.00
RC-100-1	40	1	0.997	11	980	2	0	2	9	26	3.85
	60	1	0.996	73	2580	81	2	110	7	15	6.67
RC1-50a-1	40	5	0.998	17	652	2	0	2	5	8	6.25
	60	5	1	1	912	1	0	1	5	5	10.00
	80	5	0.982	25	1932	9	1	12	3	6	8.33
RC1-50b-1	40	5	1	1	300	0	0	0	5	20	2.50
	60	5	0.981	7	951	4	0	5	5	11	4.55
	80	5	0.992	101	4325	198	3	334	2	9	5.56
RC1-75a-1	40	5	0.999	9	780	1	0	1	8	22	3.41
	60	5	1	1	1373	3	0	3	6	14	5.36
RC1-75b-1	40	5	1	1	531	0	0	0	8	25	3.00
	60	5	1	5	1344	4	0	4	6	15	5.00
RC-100-1	40	5	0.999	13	1034	2	0	2	9	28	3.57
	60	5	0.998	33	4175	96	3	122	6	18	5.56
RC1-50a-2	40	1	0.994	15	455	1	0	1	5	12	4.17
	60	1	1	1	744	1	0	1	5	5	10.00
	80	1	1	1	1595	6	0	7	4	5	10.00
RC1-50b-2	40	1	0.994	5	373	0	0	0	7	17	2.94
	60	1	0.997	5	1109	3	0	4	4	10	5.00
	80	1	0.989	27	3630	462	2	511	3	7	7.14
RC1-75a-2	40	1	0.994	19	598	2	0	2	7	25	3.00
	60	1	1	1	1158	3	0	3	6	13	5.77
RC1-75b-2	40	1	0.999	3	531	0	0	1	9	24	3.13
	60	1	0.994	29	1740	15	1	22	6	16	4.69
RC1-100-2	40	1	0.997	9	846	1	0	2	9	30	3.33
	60	1	0.998	5	2072	21	1	22	6	16	6.25
RC1-50a-2	40	5	0.998	15	455	1	0	1	5	12	4.17
	60	5	1	1	759	1	0	1	5	7	7.14
	80	5	1	1	1614	3	0	3	3	6	8.33
RC1-50b-2	40	5	1	1	362	0	0	0	6	20	2.50
	60	5	0.998	7	1145	2	0	2	4	12	4.17
	80	5	0.995	35	3991	582	3	640	3	7	7.14
RC1-75a-2	40	5	0.998	23	610	1	0	2	8	29	2.59
	60	5	1	1	1317	3	0	3	5	17	4.41
RC1-75b-2	40	5	1	5	533	0	0	0	8	28	2.68
	60	5	0.995	167	3048	46	4	222	5	18	4.17
RC1-100-2	40	5	0.999	11	837	2	0	2	8	34	2.94
	60	5	0.998	89	2804	127	3	229	6	19	5.26

Table 5: Branch-and-Price Results for RC1 Problem Instances

References

Ahuja, R., T.L. Magnanti, J.B. Orlin. 1993. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, Englewood Cliffs, New Jersey.

Barnhart, C., E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, P.H. Vance. 1998. Branch-and-price: Column generation for solving huge integer programs. *Operations Ressearch* **46** 316–329.

Berger, R.T. 1997. Location-routing models for distribution system design. Ph.D. thesis, Northwestern University, Evanston, IL.

Burlett, L. 2002. Regional operating expenses: A case study. *Southern Business and Development* URL `http://www.sb-d.com/issues/winter2002/features/save.asp`.

Christofides, N., S. Eilon. 1969. Expected distances in distribution problems. *Operational Research Quarterly* **20** 437–443.

Cornuejols, G., G.L. Nemhauser, L.A. Wolsey. 1990. The uncapacitated facility location problem. P. Mirchandani, R.L. Francis, eds., *Discrete Location Theory*. Wiley, New York, 119–171.

Desrochers, M., J. Desrosiers, M. Solomon. 1992. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research* **40** 342–354.

Desrochers, M., F. Soumis. 1989. A column generation approach to the urban transit crew scheduling problem. *Transportation Science* **23** 1–13.

Desrosiers, J., F. Soumis, M. Desrochers. 1984. Routing with time windows by column generation. *Networks* **14** 545–565.

Dumas, Y., J. Desrosiers, F. Soumis. 1991. The pickup and delivery problem with time windows. *European Journal of Operations Research* **54** 7–22.

Feillet, D., P. Dejax, M. Gendreau, C. Gueguen. 2004. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks* **44** 216–229.

Gelinas, S., M. Desrochers, J. Desrosiers, M.M. Solomon. 1995. A new branching strategy for time constrained routing problems with application to backhauling. *Annals of Operations Research* **61** 91–109.

Gorr, W.P., M.P. Johnson, S. Roehrig. 2001. Facility location model for home-delivered services: Application to the meals on wheels program. *Journal of Geographic Systems* **3** 181–197.

Hoffman, K.L., M. Padberg. 1993. Solving airline crew scheduling problems by branch-and-cut. *Management Science* **39** 657–682.

Irnich, S., D. Villeneuve. 2005. The shortest path problem with resource constraints and k-cycle elimination for $k \geq 3$. *INFORMS Journal on Computing* **to appear**.

Kohl, N., J. Desorsiers, O.B.G. Madsen, M.M. Solomon. 1999. 2-path cuts for the vehicle routing problem with time windows. *Transportation Science* **33** 101–116.

Laporte, G. 1986. Generalized subtour elimination constraints and connectivity constraints. *Journal of the Operational Research Society* **37** 509–514.

Laporte, G. 1988. Location-routing problems. B.L. Golden, A.A. Assad, eds., *Vehicle Routing: Methods and Studies*. North-Holland, Amsterdam, 163–198.

Laporte, G., Y. Nobert. 1981. An exact algorithm for minimizing routing and operating costs in depot location. *European Journal of Operational Research* **6** 224–226.

Laporte, G., Y. Nobert, D. Arpin. 1986. An exact algorithm for solving a capacitated location-routing problem. *Annals of Operations Research* **6** 293–310.

Laporte, G., Y. Nobert, P. Pelletier. 1983. Hamiltonian location problems. *European Journal of Operational Research* **12** 82–89.

Laporte, G., Y. Nobert, S. Taillefer. 1988. Solving a family of multi-depot vehicle routing and location-routing problems. *Transportation Science* **22** 161–172.

Miliotis, P. 1976. Integer programming approaches to the travelling salesman problem. *Mathematical Programming* **10** 367–378.

Miliotis, P. 1978. Using cutting planes to solve the symmetric travelling salesman problem. *Mathematical Programming* **15** 177–188.

Min, H., V. Jayaraman, R. Srivastava. 1998. Combined location-routing problems: A synthesis and future research directions. *European Journal of Operational Research* **108** 1–15.

Paltrow, S. J. 2003. Selling strategies – profiting from impatience. *The Wall Street Journal* URL `http://www.ensenda.com/about/gfx/pr_25.pdf`.

Ryan, D.M., B.A. Foster. 1981. An integer programming approach to scheduling. A. Wren, ed., *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling*. North-Holland, Amsterdam, 269–280.

Savelsbergh, M. 1997. A branch-and-price algorithm for the generalized assignment problem. *Operations Research* **45** 831–841.

Savelsbergh, M., M. Sol. 1998. Drive: Dynamic routing of independent vehicles. *Operations Research* **46** 474–490.

Savelsbergh, M.W.P., G.L. Nemhauser. 1996. Functional description of minto, a mixed integer optimizer. Tech. rep., Georgia Institute of Technology, Atlanta, Georgia.

Solomon, M.M. 1987. Algorithm for the vehicle routing and scheduling problem with time window constraints. *Operations Research* **35** 254–265.

Vance, P.H. 1993. Crew scheduling, cutting stock and column generation: Solving huge integer programs. Ph.D. thesis, Georgia Institute of Technology, Atlanta, GA.

Webb, M.H.J. 1968. Cost functions in the location of depots for multiple-delivery journeys. *Operational Research Quarterly* **19** 311–320.