IX. LOCATION ROUTING PROBLEMS

Annals of Operations Research 6(1986)293–310

AN EXACT ALGORITHM FOR SOLVING A CAPACITATED LOCATION-ROUTING PROBLEM

G. LAPORTE*, Y. NOBERT☆ and D. ARPIN*

**Ecole des Hautes Etudes Commerciales de Montréal, 5255 Ave. Decelles, Montréal H3T 1V6, Canada*

☆Départment des Sciences Administratives, Université du Québec à Montréal, 1495 Rue St-Denis, Montréal H3C 3P8, Canada

Abstract

In location-routing problems, the objective is to locate one or many depots within a set of sites (representing customer locations or cities) and to construct delivery routes from the selected depot or depots to the remaining sites at least system cost. The objective function is the sum of depot operating costs, vehicle acquisition costs and routing costs. This paper considers one such problem in which a weight is assigned to each site and where sites are to be visited by vehicles having a given capacity. The solution must be such that the sum of the weights of sites visited on any given route does not exceed the capacity of the visiting vehicle. The formulation of an integer linear program for this problem involves degree constraints, generalized subtour elimination constraints, and chain barring constraints. An exact algorithm, using initial relaxation of most of the problem constraints, is presented which is capable of solving problems with up to twenty sites within a reasonable number of iterations.

Keywords and phrases

Capacitated location-routing, integer programming, algorithm, least cost.

1. Introduction

The design of distribution systems frequently involves selecting sites on a network for a number of depots and establishing delivery routes from the depots to sets of other sites. In such contexts, location and routing decisions are closely interrelated and can not be made separately without running the risk of arriving at a suboptimal solution.

The sites can represent service locations, customer locations, or even cities, depending on the level of resolution. Location-routing problems are frequently encountered in practice. Several interesting applications are described in the surveys by Madsen [15], Perl [21] and Laporte and Nobert [11]. These are related to the food industry [24], store delivery planning [23], optimal location of regional blood banks [20], the rubber industry in Malaysia [18], the optimal location of NATO air bases [16], etc.

Solution methods for these problems are at an early stage of development when compared with the number of available algorithms for pure location or pure routing problems (see [4,5,6] for surveys of location problems and [1] for a survey of routing problems). This is due in large part to the difficulty of these problems, pointed out by some researchers ([22, p. 248], [4, p. 189]). Most algorithms for location-routing problems are approximate. The heuristic approach which is most often used (see [21,23,24]) consists of initially allocating customers to depots and of then constructing delivery routes. These location and routing decisions are then changed if a modification leads to an overall cost reduction. The process ends when no marginal improvement can be achieved by further modification.

The authors are aware of only two papers reporting the development of exact algorithms for location-routing problems. Laporte and Nobert [9] consider the problem of simultaneously locating one depot among n sites and of establishing m delivery routes from the depot to the remaining $n - 1$ sites. The problem is formulated as an integer linear program (ILP) which is solved by a constraint relaxation method; integrality is obtained by branch and bound. The largest problem solved contains fifty sites. In a more recent paper, Laporte et al. [14] study more general location-routing situations involving simultaneous choice of several depots among n sites and the optimal routing of vehicles through the remaining sites. As in Laporte and Nobert [9], the problems are formulated as ILPs which are handled by a constraint relaxation method. In Laporte et al. [14], however, integrality is reached through the gradual introduction of Gomory cutting planes (see [3] and [7]). In a first series of problems, each site may be visited only once and no route may connect two depots. The difficulty of the problem is shown to be strongly related to the nature of the distance matrix and to other factors such as the absence or presence of fixed costs on the depots and the existence of an upper bound on the number of depots which may be used. In a second series of problems, multiple visits to the same site are allowed but no fixed costs or upper bounds on the number of depots are imposed. The authors show that this problem is relatively easy to solve and constitutes in fact a relaxation of the problem of finding the shortest complete cycle in a graph, as previously analyzed by Miliotis et al. [17].

This paper considers a situation which is at the same time more general and more realistic than those studied by Laporte and Nobert [9] and by Laporte et al. [14]. A situation is considered in which the following conditions exist:

(i) the set of potential depots is disjoint from the set of sites;

(ii) multiple passages through the same site are allowed if they result in distance savings;

(iii) a non-negative weight may be assigned to each site and the total weight of each route may never exceed vehicle capacity.

It is necessary at this stage to introduce some notation. Let $G = (N, E, C)$ be a graph, where $N = \{1, \ldots, n\}$ is a set of nodes (representing the sites), where E is a set of undirected edges (i, j), and where $C = (c_{ij})$ is the symmetrical matrix of least cost routes associated with the edges C and therefore always satisfies the triangle inequality (i.e. $c_{ij} \leqslant c_{ik} + c_{kj}$ for all $i, j, k \in N$). Note that (i, j) and c_{ij} are only defined if $i < j$. Let $R \subseteq N$ be a set of *potential depots*. The number P of such nodes used as depots in the optimal solution must lie between two prespecified bounds $\underline{P} \geqslant 1$ and $\overline{P} \leqslant |R|$. The cost of using node r as a depot is equal to g_r. There are m_r identical vehicles based at depot r, each with the same capacity D but with different fixed costs f_r. To each node of $N - R$, that is, the nodes that were not potential depot sites, we associate a non-negative service requirement d_i ($\leqslant D$). Note that if any of the potential depot sites r are not used as depots, they need not be considered further. The potential depot sites have no service requirement like the nodes of $N - R$. Also note that since $d_i \leqslant D$, there will never be a need for a node to be visited by more than one vehicle to satisfy its service requirement.

The problem consists of selecting depot sites (when $\underline{P} < |R|$), of determining how many vehicles are based at each selected depot, and of establishing vehicle routes in such a way that

(i) each route starts and ends at the same depot;
(ii) all of the service requirement at a node is met by only one vehicle (the same site i may be visited more than once if this saves distance, but only one of the vehicle visits is used to meet the service requirement of the node);
(iii) the sum of all requirements satisfied by any vehicle does not exceed D;
(iv) the number of nodes used as depots lies between two prespecified bounds $\underline{P}$ and $\overline{P}$: $1 \leqslant \underline{P} \leqslant P \leqslant \overline{P} \leqslant |R|$;
(v) for each node r used as a depot, the number of vehicles lies between two prespecified bounds $\underline{m}_r$ and $\overline{m}_r$: $1 \leqslant \underline{m}_r \leqslant m_r \leqslant \overline{m}_r$;
(vi) the total cost is minimized.

This problem is a generalization of many well-known problems studied in the operational research literature. When $P = |R|$, and according to the value of $|R|$, it reduces to either the travelling salesman problem [2], the multiple travelling salesman problem [8], the capacitated vehicle routing problems [10], or a family of multi-depot vehicle routing problems recently treated by the authors [12].

Our purpose is to provide an ILP formulation and an exact algorithm for this problem.

2. Formulation

In addition to the notation already introduced, we define

S : a subset of $N-R$.

L : an arbitrarily large number.

$$\ulcorner t \urcorner = \begin{cases} \text{the smallest integer greater than or equal to } t \text{ if } t > 0 \\ 1 \text{ otherwise.} \end{cases}$$

x_{ij} : a variable indicating the number of times edge (i, j) is used in the optimal solution. x_{ij} is not defined if $i \geqslant j$, if $i, j \in R$ or if $d_i + d_j > D$. x_{ij} must be interpreted as x_{ji} whenever $i > j$.

y_r : a binary variable indicating whether node r is used as a depot ($y_r = 1$) or not ($y_r = 0$).

z : the total system cost.

The problem can be formulated as follows:

(P) $$\text{minimize } z = \sum_{i,j \in N} c_{ij} x_{ij} + \sum_{r \in R} (g_r y_r + f_r m_r)$$

subject to

$$\sum_{i<k} x_{ik} + \sum_{k<j} x_{kj} = 2 \qquad (k \in N-R) \qquad (1)$$

$$\sum_{i<r} x_{ir} + \sum_{r<j} x_{rj} = 2m_r \qquad (r \in R) \qquad (2)$$

$$\sum_{i,j \in S} x_{ij} \leqslant |S| - \left\ulcorner \frac{\sum_{k \in S} d_k}{D} \right\urcorner \qquad (S \subseteq N-R,\ |S| \geqslant 3) \qquad (3)$$

$$x_{i_1 i_2} + 3x_{i_2 i_3} + x_{i_3 i_4} \leqslant 4 \qquad (i_1, i_4 \in R;\ i_2, i_3 \in N-R) \qquad (4)$$

$$x_{i_1 i_2} + x_{i_{h-1} i_h} + 2 \sum_{i,j \in \{i_2, \ldots, i_{h-1}\}} x_{ij} \leqslant 2h-5$$

$$(h \geqslant 5;\ i_1, i_h \in R;$$
$$i_2, \ldots, i_{h-1} \in N-R; \qquad (5)$$

$$y_r \leqslant m_r \leqslant L y_r \qquad (r \in R) \qquad (6)$$

$$\underline{m}_r \leqslant m_r \leqslant \bar{m}_r \qquad (r \in R) \qquad (7)$$

$$\underline{P} \leqslant \sum_{r \in R} y_r \leqslant \bar{P} \qquad (8)$$

$$y_r = 0, 1 \qquad (r \in R) \qquad (9)$$

$$x_{ij} = \begin{cases} 0, 1 & (i, j \in N - R) \\ 0, 1, 2 & (i \text{ or } j \in R) \end{cases} \qquad (10)$$

In this formulation, the objective function is defined as the sum of travel costs, depot operating costs and vehicle costs. Constraint (2) specifies that each site not used as a depot must be *serviced* exactly once by any given vehicle. Of course, any site may be *visited* more than once if this saves distance, but this need not appear explicitly in the formulation. For example, if site k lies on two shortest paths (between i_1 and j_1 and between i_2 and j_2, say), then it may be visited twice, but only once will a service be provided. Similarly, constraint (2) expresses the fact that m_r vehicles must leave and enter each depot in R. Constraint (3) ensures that the solution does not contain any illegal subtours, i.e. subtours disjoint from R or subtours having a total weight exceeding D. These constraints generalize the *subtour elimination constraints* used by Dantzig et al. [2] for the travelling salesman problem; their derivation is provided in [10] and [19, p. 48]. Constraints (4) and (5) are *chain barring constraints.* They ensure that each route starts and ends at the same depot. As their development is relatively lengthy, it is explained in the next section. Constraint (6) means that no vehicle can be based at a node which is not used at a depot; moreover, if a node is used as a depot, it must have at least one vehicle assigned to it. Constraint (7) indicates that the number of vehicles assigned to any given depot must lie between prespecified bounds. Similarly, constraint (8) expresses the fact that the total number of nodes used as depots must lie between two bounds. Constraint (9) is used to determine which potential depot sites are used in the optimal solution. Finally, constraint (10) indicates how many times each edge (i, j) is used in the optimal solution; $x_{ij} = 2$ corresponds to a return trip between i and j.

3. Chain barring constraints

Consider a relaxed problem (P') obtained from (P) by removing constraints (4) and (5). The optimal solution to (P') may contain *chains* between two depots, i.e. sequences of nodes $(i_1, \ldots, i_h)$ where

(i) $i_1, i_h \in R;\ i_2, \ldots, i_{h-1} \in N-R;\ i_1 \neq i_h;$

(ii) $x_{i_t i_{t+1}} = 1 \quad (t = 1, \ldots, h-1).$

We wish to generate constraints that will eliminate such chains without cutting off any feasible solution to (P). This will be accomplished for chains of various sizes.

3.1. CHAINS INVOLVING AT LEAST 4 NODES

First assume that $h \geqslant 4$. Consider a chain $H = (i_1, i_2, \ldots, i_h)$. For $\ell > k$, we define $S_{k\ell} = \{i_k, i_{k+1}, \ldots, i_\ell\}$. Let $X = x_{i_1 i_2}$, $Y = x_{i_{h-1} i_h}$ and $Z = \Sigma_{i,j \in S_{2,h-1}} x_{ij}$ and let $\bar{X}$, $\bar{Y}$, $\bar{Z}$ be the values taken by X, Y, Z, respectively. In a feasible solution, $\bar{X}$, $\bar{Y} \in \{0, 1, 2\}$ and $\bar{Z}$ is an integer in $[0, u(\bar{X}, \bar{Y})]$, where $u(\bar{X}, \bar{Y})$ is an upper bound on Z whose value depends on $\bar{X}$ and $\bar{Y}$. In the following development, it is assumed that the $u(\bar{X}, \bar{Y})$s are known constants. Later in this section, we indicate how to determine the smallest bound for given $\bar{X}$ and $\bar{Y}$. Let

$$Q = \{(\bar{X}, \bar{Y}, \bar{Z}) : \bar{X}, \bar{Y} \in \{0,1,2\}, \bar{Z} \in [0, u(\bar{X}, \bar{Y})], \bar{Z} \text{ integer}\}. \tag{11}$$

A valid cut is a constraint of the form

$$aX + bY + Z \leqslant d \tag{12}$$

(where $a, b \in \mathbb{R}$ and $d \geqslant 0$), which satisfies

$$d - a\bar{X} - b\bar{Y} \geqslant u(\bar{X}, \bar{Y}) \qquad (\bar{X}, \bar{Y} \in \{0,1,2\}) \tag{13}$$

while cutting the current solution characterized by $\bar{X} = 1$, $\bar{Y} = 1$ and $\bar{Z} = h - 3$.

We seek the constraint which provides the deepest cut, i.e. the constraint that minimizes $(d - a - b)$, the value taken by the left-hand side of (13) when $\bar{X} = \bar{Y} = 1$. The coefficients of the cut are given by the solution of $(P^\star)$:

$(P^\star)$ minimize $d - a - b$

subject to $$d - a\bar{X} - b\bar{Y} \geqslant u(\bar{X}, \bar{Y}) \qquad (\bar{X}, \bar{Y} \in \{0,1,2\}) \tag{14}$$

$$a, b \text{ free},\ d \geqslant 0. \tag{15}$$

The optimal values of a, b and d can be obtained by means of the simplex method or by using a physical model. When (P^*) possesses an infinite number of optimal solutions, we retain only the constraints which define facets of Q. Note that the cut is effective only if it eliminates the current infeasible solution, i.e. if

$$d - a - b < h - 3. \tag{16}$$

It is straightforward to check [constraints (4) and (5)] that this condition is always satisfied. We now proceed to the generation of chain barring constraints in some particular cases.

3.2. CHAINS INVOLVING 4 NODES

Consider a chain $H = (i_1, i_2, i_3, i_4)$. Here, $X = x_{i_1 i_2}$, $Y = x_{i_3 i_4}$ and $Z = x_{i_2 i_3}$. We first observe that in any feasible solution to (P), Z can only take a positive value if $X + Y \leqslant 1$. Otherwise, edge (i_2, i_3) can not enter the solution. The values of $u(\bar{X}, \bar{Y})$ are therefore

$$u(\bar{X}, \bar{Y}) = \begin{cases} 1 & \text{if } \bar{X} + \bar{Y} \leqslant 1 \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

It is easy to verify that the optimal cutting plane (12) is uniquely determined by the following three points of Q: (0, 1, 1), (1, 0, 1) and (2, 2, 0). This plane can be expressed in terms of the x_{ij}'s as

$$x_{i_1 i_2} + 3x_{i_2 i_3} + x_{i_3 i_4} \leqslant 4, \tag{18}$$

i.e. constraint (4).

3.3. CHAINS INVOLVING AT LEAST 5 NODES

Now consider a chain $H = (i_1, i_2, \ldots, i_h)$, where $h \geqslant 5$. As above, $X = x_{i_1 i_2}$, $Y = x_{i_{h-1} i_h}$ and $Z = \Sigma_{i,j \in S_{2,h-1}} x_{ij}$. Since constraint (3) is satisfied, it is feasible to use only one vehicle to service all nodes of $S_{2,h-1}$ in the optimal solution (otherwise the current values of X and Y would exceed 1).

We will first illustrate the computation of $u(\bar{X}, \bar{Y})$ by taking a special case. Let $\bar{X} = 2$ and $\bar{Y} \neq 2$; any feasible solution for (P) must satisfy

$$\sum_{j \in S_{3,h-1}} x_{i_2 j} = 0 \tag{19}$$

and therefore

$$Z = \sum_{i,j \in S_{2,h-1}} x_{ij} = \sum_{i,j \in S_{3,h-1}} x_{ij} \leqslant (h-4) = u(\bar{X}, \bar{Y}). \tag{20}$$

By applying the same kind of reasoning, it is possible to obtain bounds $u(\bar{X}, \bar{Y})$ on Z for all combinations of $\bar{X}$ and $\bar{Y}$:

$$u(\bar{X}, \bar{Y}) = \begin{cases} h-5 & \text{if } \bar{X} + \bar{Y} = 4 \\ h-4 & \text{if } 2 \leqslant \bar{X} + \bar{Y} \leqslant 3 \\ h-3 & \text{if } \bar{X} + \bar{Y} \leqslant 1 . \end{cases} \tag{21}$$

In order to find a chain barring constraint, it suffices to substitute in $(P^\star)$ $\bar{X}$, $\bar{Y}$ and $u(\bar{X}, \bar{Y})$ by their values given by (21), to make the change of variable $d' = d - h + 5$ and to solve the program.

The resulting constraint is given by

$$X + Y + 2Z \leqslant 2h - 5, \tag{22}$$

i.e. constraint (5).

The two constraints described by (18) and (22) constitute facets of Q and provide the deepest cut.

3.4. CHAINS INVOLVING LESS THAN 4 NODES

The only chains left out of the above analysis are those involving 2 or 3 nodes. Illegal chains containing only 2 nodes can not occur since x_{ij} is not defined when i and $j \in R$. There is only one type of solution which could theoretically contain chains involving 3 nodes after all other types of chains have been eliminated: these would occur in subtours made up of an alternance of nodes from R and of nodes from $N - R$. Let $(i_1, j_1, i_2, j_2, \ldots, i_\ell, j_\ell, i_1)$ be such a subtour where $i_1, i_2, \ldots, i_\ell \in R$ and $j_1, j_2, \ldots, j_\ell \in N - R$. However, we need not consider this case since it is never disadvantageous to replace this subtour by ℓ subtours of the form $(i_k^\star, i_k, i_k^\star)$, where

$$i_k^\star = \begin{cases} j_k & \text{if } c_{i_k j_k} \leqslant c_{i_k j_{k-1}} \qquad (j_0 = j_\ell) \\ j_{k-1} & \text{otherwise,} \end{cases} \tag{23}$$

since

$$2c_{i_k i_k^*} \leqslant c_{i_k j_{k-1}} + c_{i_k j_k} . \tag{24}$$

4. Algorithm

The algorithm developed to solve (P) can be summarized as follows.

Step 0: Obtain a first feasible solution by means of an appropriate heuristic. Let z^* be the total system cost of that solution.

Step 1: Select a subproblem from the list. (The first subproblem will include constraints (1), (2), (6), (7) and (8) of (P) as well as the upper bounds on the variables.)

Step 2: Solve the subproblem using the simplex method. (We used the Land–Powell code [7].) Let $\bar{z}$ be the cost of the least cost solution to the subproblem.

Step 3: If $\bar{z} \geqslant z^*$, proceed to step 9.

Step 4: The current solution contains

(i) sets of nodes $\{i_1, \ldots, i_\ell\}$ $(\ell > 1)$ corresponding to chains $(i_1, \ldots, i_\ell)$ such that $\{i_2, \ldots, i_{\ell-1}\} \cap R = \phi$ if $\ell > 2$ and for which *all* variables $x_{i_1 i_2}, x_{i_2 i_3}, \ldots, x_{i_{\ell-1} i_\ell}$ have been previously *fixed* at 1, and

(ii) nodes of $N - R$ not belonging to such chains (we define for each such node i a set $\{i\}$).

For convenience, we refer to these sets of nodes S_k (corresponding to chains or single nodes) as *components.* Each S_k has an associated weight $w(S_k)$ defined as

$$w(S_k) = \sum_{i \in S_k} d_i . \tag{25}$$

Consider a component S_k. If it corresponds to a chain, let p_k and q_k be the end nodes of that chain; if it corresponds to a node i, let $p_k = q_k = i$. In the first case, $x_{p_k q_k}$ can be *forced* to zero as long as $p_k, q_k \in N - R$. Now consider two components S_r and S_s and let $i \in \{p_r, q_r\}$, $j \in \{p_s, q_s\}$. Then x_{ij} can be forced to zero if $i, j \in N - R$ and $w(S_r) + w(S_s) > D$.

If step 4 has resulted in forcing any variable to zero, proceed to step 2.

Step 5: Check whether the current solution contains illegal subtours (i.e. subtours disconnected from R or having a total weight exceeding D). If there are no illegal subtours, proceed to step 6. Otherwise, generate a type (3) constraint for each illegal subtour and proceed to step 2.

Step 6: Check whether the current solution contains illegal chains between depots (in this context, a depot is a node r of R for which $y_r = 1$). If there are no illegal chains, proceed to step 7. Otherwise, generate for each chain a subtour elimination constraint (3) or a chain barring constraint [(4) or (5)] and proceed to step 2.

Step 7: If the solution is integer, store it and set $z^* = \bar{z}$; proceed to step 9. Otherwise, execute the next step.

Step 8: Select a fractional variable to branch upon and create new branches in the search tree according to the procedure described in [7]. Go to step 10.

Step 9: Back up in the search tree. This consists of modifying the level of the search tree (from λ' to λ, where $\lambda' \geqslant \lambda$) by *freeing* all variables either forced to zero or fixed at some integer value at levels $\lambda, \lambda + 1, \ldots, \lambda'$. Here we used the BACKUP subroutine of the Land–Powell code [7].

Step 10: Update the list of subproblems. If the list is empty, terminate and print the best solution. Otherwise, proceed to step 1.

5. Computational results

The algorithm was tested on a number of randomly generated problems for various parameter choices. In all problems, the c_{ij}'s were defined as the straight line distance between points (X_i, Y_i) and (X_j, Y_j) generated according to a uniform distribution on $[0, 100]^2$. We then carried out three series of tests.

Test series 1: Problems involving no capacity restrictions, no vehicle costs and no fixed costs on the depots. In these problems, constraints (7) were removed and $\underline{P}$ was set equal to 1 (see table 1).

Test series 2: Weights d_i were generated according to a uniform distribution on [0, 100]. As in previous studies [10,13], the vehicle capacity D was defined as

$$D = (1 - \alpha) \max_{i \in N-R} \{d_i\} + \alpha \sum_{i \in N-R} d_i \tag{26}$$

when α is a parameter chosen in [0, 1]. The remaining parameters were selected as in test series 1. Note that fixing α at 1 is equivalent to removing the capacity restriction. Setting α at a low value results in tight and relatively hard to solve problems (see table 2).

Test series 3: Problems were generated as in test series 2, but then the effect of fixed costs on vehicles and on depots was studied. These two types of fixed costs were never used simultaneously so that the effect of each type could be isolated (see table 3).

Table 1

Problems without capacity restrictions or fixed costs

n	$\|R\|$	$\bar{P}$	Problem no.	Time (secs)	No. of simplex iterations	Max no. of effective constraints	No. of subtour elimination constraints	No. of chain barring constraints
10	4	3	1	14.13	799	24	11	0
			2	2.02	112	22	10	0
			3	0.97	66	20	1	0
15	6	4	1	2.50	91	29	1	0
			2	35.72	1050	34	11	0
			3	8.35	239	33	6	0
20	8	5	1	44.43	827	41	5	0
			2	22.47	368	44	9	4
			3	52.55	1056	45	12	0
25	10	6	1	> 300	2974	52	13	5
			2	> 300	2828	52	13	5
			3	38.54	492	48	3	7

Table 2

Problems with capacity restrictions and without fixed costs

n	$\|R\|$	$\bar{P}$	α	Problem no.	Time (secs)	No. of simplex iterations	Max no. of effective constraints	No. of subtour elimination constraints	No. of chain barring constraints
10	4	3	0.00	1	17.75	837	28	30	14
				2	1.05	72	20	6	0
				3	24.66	1025	26	20	16
			0.25	1	57.94	2044	30	22	12
				2	> 300	10 349	31	44	25
				3	5.97	326	26	12	4
			0.50	1	7.84	450	24	14	0
				2	134.48	4808	27	16	43
				3	14.61	710	25	8	16
			0.75	1	9.92	552	23	12	0
				2	19.71	1082	24	9	18
				3	1.66	102	22	5	0
			1.00	1	9.73	552	23	12	0
				2	4.51	318	22	4	0
				3	1.85	104	22	5	0
15	6	4	0.00	1	> 300	6231	44	56	20
				2	104.27	2146	44	46	45
				3	> 300	6552	50	40	41
			0.25	1	> 300	5338	39	34	17
				2	> 300	5219	38	33	12
				3	150.91	3072	34	25	12
			0.50	1	51.43	1139	34	23	7
				2	12.85	331	37	16	7
				3	11.70	353	30	7	7
			0.75	1	129.82	2570	37	19	30
				2	3.98	119	30	10	2
				3	2.56	92	26	3	3
			1.00	1	3.52	121	26	4	0
				2	2.11	76	27	6	0
				3	2.57	92	26	3	3

Table 2 (continued)

n	$\|R\|$	$\bar{P}$	α	Problem no.	Time (secs)	No. of simplex iterations	Max no. of effective constraints	No. of subtour elimination constraints	No. of chain barring constraints
20	8	5	0.00	1	> 300	4163	63	72	22
				2	> 300	4239	55	59	20
				3	> 300	3352	55	59	26
			0.25	1	> 300	3058	49	23	19
				2	> 300	4600	49	46	66
				3	> 300	3214	53	47	4
			0.50	1	> 300	4244	45	8	24
				2	> 300	4534	45	14	34
				3	> 300	4120	48	42	34
			0.75	1	298.40	2852	46	6	6
				2	44.13	762	37	6	7
				3	217.67	3470	44	14	6
			1.00	1	297.3	2852	46	6	6
				2	43.18	762	37	6	7
				3	97.50	1817	39	5	0

Table 3

Problems with capacity restrictions and fixed costs

n	$\|R\|$	$\bar{P}$	α	f_r (vehicle costs)	g_r (depot costs)	Problem no.	Time (secs)	No. of simplex iterations	Max no. of effective constraints	No. of subtour elimination constraints	No. of chain barring constraints
10	4	3	0.50	0	0	1	7.73	450	24	14	0
						2	133.51	4808	27	16	43
						3	4.32	710	25	8	16
				50	0	1	5.39	292	23	14	0
						2	15.17	649	28	16	19
						3	3.56	174	24	6	5
				100	0	1	6.09	305	25	20	0
						2	22.68	959	28	24	27
						3	6.04	311	26	6	9
				200	0	1	6.06	305	25	20	0
						2	41.23	1509	29	20	24
						3	5.85	283	25	6	9
				0	50	1	6.25	383	27	16	0
						2	27.08	1140	27	25	15
						3	3.64	196	23	9	0
				0	100	1	5.02	291	28	17	0
						2	5.09	269	25	19	0
						3	4.79	258	23	9	0
				0	200	1	4.05	247	26	5	0
						2	5.36	258	26	20	0
						3	6.30	341	23	8	0

Table 3 (continued)

n	$\|R\|$	$\bar{P}$	α	f_r (vehicle costs)	g_r (depot costs)	Problem no.	Time (secs)	No. of simplex iterations	Max no. of effective constraints	No. of subtour elimination constraints	No. of chain barring constraints
15	6	4	0.50	0	0	1	51.67	1139	34	23	7
						2	13.06	331	37	16	7
						3	11.63	353	30	7	7
				50	0	1	17.73	428	34	20	11
						2	147.80	2497	41	37	53
						3	156.37	2581	39	29	20
				100	0	1	17.62	396	35	18	9
						2	60.35	1142	41	23	17
						3	> 300	5125	37	50	63
				200	0	1	140.25	2272	36	34	29
						2	> 300	3527	41	47	40
						3	> 300	4583	44	58	63
				0	50	1	23.35	641	36	20	0
						2	25.97	610	38	24	0
						3	19.79	609	24	14	0
				0	100	1	22.15	585	37	19	0
						2	72.59	1094	40	34	0
						3	31.00	855	33	13	0
				0	200	1	20.13	537	37	19	0
						2	15.88	447	38	17	0
						3	28.98	891	35	14	0

Table 3 (continued)

n	$\|R\|$	$\bar{P}$	α	f_r (vehicle costs)	g_r (depot costs)	Problem no.	Time (secs)	No. of simplex iterations	Max no. of effective constraints	No. of subtour elimination constraints	No. of chain barring constraints
20	8	5	0.75	0	0	1	298.40	2852	46	6	6
						2	44.45	762	37	6	7
						3	217.96	3470	44	14	6
				50	0	1	207.85	2407	49	19	10
						2	180.29	2136	46	19	16
						3	> 300	4370	44	9	15
				100	0	1	245.80	2452	50	24	20
						2	115.80	1462	47	18	20
						3	41.79	707	41	7	6
				200	0	1	> 300	3079	49	29	38
						2	212.31	2415	49	24	26
						3	20.84	411	40	8	
				0	50	1	87.08	1331	46	27	0
						2	94.49	1278	46	21	0
						3	47.13	706	42	14	0
				0	100	1	148.11	1846	48	21	0
						2	216.88	2728	43	17	0
						3	38.60	543	44	17	0
				0	200	1	136.54	1882	45	16	0
						2	75.16	304	45	24	0
						3	46.71	647	46	22	0

In each case, three problems were attempted for various combinations of n (number of sites), $|R|$ (number of potential depot locations), and $\bar{P}$ (maximum number of depots in the solutions).

All problems were solved on the University of Montréal CYBER 855 computer, using an FTN5 compiler. Problems "failed" when they could not be solved within 300 CPU seconds. Tables 1, 2 and 3 report the results obtained.

These results indicate that problems containing up to twenty sites can be solved exactly by our algorithm within our solution time criterion. To the authors' knowledge, this is the first reported attempt in the operational research literature to provide an optimal solution to problems of such complexity and possessing the characteristics of (i) simultaneous location and routing, (ii) capacity constraints, (iii) fixed costs on vehicles or on depots, (iv) bounds on the number of depots, and (v) bounds on the number of vehicles per depot. The main factor explaining the success of the algorithm lies in the initial relaxation of most of the problem constraints. It can be observed that the maximum number of effective constraints in the course of the algorithm generally lies between $2n$ and $3n$, while the number of potential constraints is of the order of 2^n.

As observed in other studies [10,13], the difficulty of the problem is inversely related to the size of α which controls the vehicle capacity. Imposing large depot costs (see table 3) tends to produce easier problems, but the same can not be said about vehicle costs, which do not seem to affect computation times one way or the other.

6. Conclusions

We have provided an integer linear programming formulation and an exact algorithm for the solution of an important class of capacitated location-routing problems. The formulation incorporates, as do many problems of the same family, (i) degree constraints, (ii) generalized subtour elimination constraints (see [10]), and (iii) chain barring constraints.

Our results using the exact algorithm show that problems involving up to about twenty nodes can be solved optimally within a reasonable time. This appears to be the first time such complex problems have been dealt with by means of an exact algorithm.

Acknowledgements

The authors are grateful to the Canadian Natural Sciences and Engineering Research Council (Grants A4747 and A5486) and to the Québec Government (F.C.A.C. Grant 83EQ0428) for their financial support. Thanks are also due to the anonymous referees for their helpful comments.

References

[1] L.D. Bodin, B.L. Golden, A. Assad and M. Ball, Routing and scheduling of vehicles and crews. The state of the art, Computers and Operations Research 10(1983)69.
[2] G.B. Dantzig, R. Fulkerson and S.M. Johnson, Solution of a large-scale travelling salesman problem, Oper. Res. 2(1954)393.
[3] R. Gomory, An algorithm for integer solutions to linear programs, in: *Recent Advances in Mathematical Programming* (McGraw-Hill, 1963) p. 269.
[4] Y.G. Handler and P.B. Mirchandani, *Location on Networks. Theory and Algorithms* (MIT Press, Cambridge, MA, 1979).
[5] J. Krarup and T.M. Pruzan, Selected families of location problems, Ann. Disc. Math. 5(1979) 327.
[6] J. Krarup and T.M. Pruzan, The simple plant location problem: Survey and synthesis, Eur. J. Oper. Res. 12(1983)36.
[7] A.H. Land and S. Powell, *FORTRAN Codes for Mathematical Programming: Linear, Quadratic and Discrete* (Wiley, 1973).
[8] G. Laporte and Y. Nobert, A cutting planes algorithm for the m-salesmen problem, J. Oper. Res. Soc. 31(1980)1017.
[9] G. Laporte and Y. Nobert, An exact algorithm for minimizing routing and operating costs in depot location, Eur. J. Oper. Res. 6(1981)224.
[10] G. Laporte and Y. Nobert, A branch and bound algorithm for the capacitated vehicle routing problem, Operations Research Spektrum 5(1983)77.
[11] G. Laporte and Y. Nobert, Les problèmes mixtes de localisation et de construction de tournées: un état de la question, Ecole des Hautes Etudes Commerciales de Montréal (1985).
[12] G. Laporte, Y. Nobert and D. Arpin, Optimal solutions to capacitated multidepot vehicle routing problems, Congressus Numerantium 44(1984)283.
[13] G. Laporte, Y. Nobert and M. Desrochers, Optimal routing under capacity and distance restrictions, Oper. Res. 33(1985)1050.
[14] G. Laporte, Y. Nobert and P. Pelletier, Hamiltonian location problems, Eur. J. Oper. Res. 12 (1983)82.
[15] O.B.G. Madsen, Methods for solving combined two-level location-routing problems of realistic dimensions, Eur. J. Oper. Res. 12(1983)295.
[16] D.R. McLain, M.L. Durchholz and W.B. Wilborn, U.S.A.F. EDSA routing and operating location selection study, Report No. XPSR84-3, Operations Research Division, Directorate of Studies and Analysis, Scott Air Force Base, Illinois (1984).
[17] P. Miliotis, G. Laporte and Y. Nobert, Computational comparison of two methods for finding the shortest complete cycle or circuit in a graph, RAIRO 15(1981)233.
[18] J. Nambiar and L.G. Chalmet, A location and vehicle scheduling problem in collecting and processing rubber, paper presented at the TIMS/ORSA Conference, New Orleans (1979).
[19] Y. Nobert, Construction d'algorithmes optimaux pour des extensions au problème du voyageur de commerce, Thèse de doctorat, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal (1982)
[20] I. Or and W.P. Pierskalla, A transportation location-allocation model for regional blood banking, AIIE Trans. 2(1979)86.
[21] J. Perl, A unified warehouse location-routing problem, Ph.D. Dissertation, Department of Civil Engineering, Northwestern University, Evanston, Illinois (1983).
[22] G.K. Rand, Methodological choices in depot location studies, Operational Research Quarterly 27(1976)241.
[23] TDF (Transportforskningsdelegationen), Distribution planning using mathematical methods, Report 1977: 1, Contract Research Group for Applied Mathematics, Royal Institute of Technology, Sweden (1977).
[24] C.D.T. Watson-Gandy and P.J. Dorn, Depot location with van salesmen. A practical approach, Omega 1(1973)321.