DISCRETE
APPLIED
MATHEMATICS

ELSEVIER Discrete Applied Mathematics 50 (1994) 239–254

Vehicle routing with split deliveries

Moshe Dror[a,b], Gilbert Laporte[b,*], Pierre Trudeau[b,c]

[a] *Decision Sciences, College of Business, University of Arizona, Tucson, AZ 85721, USA*
[b] *Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale A, Montréal, Québec, Canada H3C 3J7*
[c] *Ad Opt, 4475 boulevard Saint-Laurent, Montréal, Québec, Canada H2W 1Z8*

Received 2 March 1990; revised 1 May 1992

Abstract

This paper considers a relaxation of the classical vehicle routing problem (VRP), in which split deliveries are allowed. As the classical VRP, this problem is NP-hard, but nonetheless it seems more difficult to solve exactly. It is first formulated as an integer linear program. Several new classes of valid constraints are derived, and a hierarchy between these is established. A constraint relaxation branch and bound algorithm for the problem is then described. Computational results indicate that by using an appropriate combination of constraints, the gap between the lower and upper bounds at the root of the search tree can be reduced considerably. These results also confirm the quality of a previously published heuristic for this problem.

Key words: Split delivery vehicle routing problem; Subtour elimination constraints; Connectivity constraints; k-split cycles; Fractional cycle elimination constraints

1. Introduction

The classical *Vehicle Routing Problem* (VRP) can be defined as follows. Let $G = (N, A)$ be a graph where $N = \{0, \ldots, n\}$ is a set of vertices corresponding to cities, and $A = \{(i, j): i, j \in N, i \neq j\}$ is the arc set. Vertex 0 represents a *depot* at which a fleet of m vehicles is based; the remaining vertices correspond to *customers*. In general, m belongs to some interval $[\underline{m}, \bar{m}]$, where $1 \leqslant \underline{m} \leqslant \bar{m} \leqslant n$. Vehicles may have equal or different capacities. Let vehicle v have a capacity equal to Q_v. Every vertex i of $N \setminus \{0\}$ has a nonnegative demand $q_i \leqslant \max_v \{Q_v\}$ and every arc (i, j) has an associated nonnegative distance or travel cost c_{ij}. The VRP consists in determining a set

* Corresponding author.

SSDI 0166-218X(92)00172-C

of minimum cost vehicle routes:

(i) starting and ending at the depot;

(ii) such that every customer is visited once by one vehicle and

(iii) such that the total demand of any route does not exceed the capacity of the vehicle assigned to the route.

It is well known that the VRP is NP-hard since, when $m = 1$ and $Q_1 \geqslant \sum_{i=1}^{n} q_i$, it then reduces to the *Travelling Salesman Problem* (TSP). There exists an abundant literature on the VRP and related problems. For recent surveys on algorithms, see [4, 13, 15]. For results on the worst-case behaviour of some heuristic algorithms, see [1, 2]. Recently, Dror and Trudeau [7, 8] have investigated a relaxation of the VRP in which condition (ii) is removed, i.e., customer demand can be split between several vehicles. In this context, it is no longer necessary to assume that $q_i \leqslant \max_v\{Q_v\}$. This variant of the VRP is called the *Split Delivery Vehicle Routing Problem* (SDVRP). Dror and Trudeau [7, 8] have proposed a heuristic algorithm for the SDVRP and have shown that allowing split deliveries can yield substantial savings, both in the total distance travelled and in the number of vehicles used in the optimal solution. Unfortunately, the SDVRP is still NP-hard [8].

The object of this paper is to prove an integer linear programming (ILP) formulation including new families of valid inequalities, as well as an exact constraint relaxation algorithm for the SDVRP. The paper is structured as follows. The ILP formulation is presented in Section 2. In Section 3, we provide a detailed discussion of subtour elimination and connectivity constraints in the context of the SDVRP. New classes of valid constraints are introduced in Section 4. The algorithm description is contained in Section 5, and computational results are reported in Section 6. The conclusion follows in Section 7.

2. Formulation

Let x_{ijv} be a binary variable defined for $i \neq j$ and equal to 1 if and only if in the optimal solution, vehicle v travels directly from i to j. Let y_{iv} be the proportion of the ith customer demand delivered by vehicle v. The problem is then:

$$\text{minimize} \quad \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{v=1}^{\bar{m}} c_{ij} x_{ijv}, \tag{1}$$

$$\text{subject to} \quad \sum_{i=0}^{n} x_{ikv} - \sum_{j=0}^{n} x_{kjv} = 0 \quad (k = 0, \ldots, n;\ v = 1, \ldots, \bar{m}), \tag{2}$$

$$\sum_{v=1}^{\bar{m}} y_{ij} = 1 \qquad (i = 1, \ldots, n), \tag{3}$$

$$\sum_{i=1}^{n} q_i y_{iv} \leqslant Q_v \qquad (v = 1, \ldots, \bar{m}), \tag{4}$$

$$\sum_{j=0}^{n} x_{ijv} \geqslant y_{iv} \qquad (i = 1, \ldots, n;\ v = 1, \ldots, \bar{m}); \tag{5}$$

subtour elimination and connectivity constraints (6)

$$x_{ijv} \in \{0,1\} \qquad (i,j = 0,\ldots,n;\ v = 1,\ldots,\bar{m}), \tag{7}$$

$$0 \leqslant y_{iv} \leqslant 1 \qquad (i = 1,\ldots,n;\ v = 1,\ldots,\bar{m}), \tag{8}$$

In this formulation, constraints (2) are flow conservation conditions. Constraints (3) specify that the demand of any customers is entirely satisfied. Constraints (4) ensure that vehicle capacities are never exceeded, while constraints (5) guarantee that if customer i is visited by vehicle v, then the same vehicle leaves that customer. *Subtour elimination constraints* (6) require a more elaborate discussion and will be described in Section 3. Note that summing up constraints (5) over all vehicles yields the following *connectivity constraints*:

$$\sum_{v=1}^{\bar{m}} \sum_{j=0}^{n} x_{ijv} \geqslant \sum_{v=1}^{\bar{m}} y_{iv} = 1 \quad (i = 1,\ldots,n), \tag{9}$$

implying that any customer i will receive at least one visit.

3. Subtour elimination and connectivity constraints

In this section, we provide valid subtour elimination and connectivity constraints for the SDVRP, and compare them with similar constraints developed for the classical VRP.

Subtour elimination constraints for the SDVRP are derived from the corresponding constraints for the TSP and for the VRP. First consider the standard subtour elimination constraints introduced by Dantzig et al. [5] for the TSP:

$$\sum_{i,j \in S} x_{ij} \leqslant |S| - 1 \quad (S \subset N\backslash\{0\};\ 2 \leqslant |S| \leqslant n-1). \tag{10}$$

These constraints eliminate all subtours defined over subsets of $N\backslash\{0\}$ containing between 2 and $n-1$ vertices. Since in the TSP there is only one vehicle, x_{ij} must be interpreted as x_{ij1} in the SDVRP formulation. It s straightforward to show that subtour elimination constraints (10) are equivalent to the following connectivity constraints:

$$\sum_{i \in S,\, j \in \bar{S}} x_{ij} \geqslant 1 \quad (S \subset N\backslash\{0\};\ 2 \leqslant |S| \leqslant n-1), \tag{11}$$

where $\bar{S} = N\backslash S$.

In the case of the classical VRP, constraints (10) can be strengthened to:

$$\sum_{v=1}^{\bar{m}} \sum_{i,j \in S} x_{ijv} \leqslant |S| - V(S) \quad (S \subseteq N\backslash\{0\};\ |S| \geqslant 2), \tag{12}$$

where $V(S)$ is the number of vehicles required to serve all nodes of S in any feasible VRP solution (see, e.g., [10, 11, 14, 16]. The value of $V(S)$ can be determined by solving a bin packing problem [10], but a lower bound is often used. In this paper constraints (12) are initially relaxed and successively introduced. At a given solution, $V(S)$ is obtained by first determining $W(S) = \{v: x_{ijv} > 0, i \in S, j \notin S\}$, and $V(S)$ is the smallest number of vehicles of $W(S)$ necessary to cover the total demand of S. Again, the equivalence of (12) with the following connectivity constraints is immediate:

$$\sum_{v=1}^{\bar{m}} \sum_{i \in S,\, j \in \bar{S}} x_{ijv} \geqslant V(S) \quad (S \subseteq N \backslash \{0\};\, |S| \geqslant 2). \tag{13}$$

Constraints (12) or (13) eliminate two types of infeasibilities: (i) subtours disconnected from the depot and (ii) vehicle routes connected to the depot, but whose total demand exceeds the vehicles capacity.

Observe that constraints (12) are invalid for the SDVRP. Indeed, consider the example in Fig. 1, with $S = \{1, 2, 3, 4, 5\}$, $Q_v = 3$ for all v, $q_1 = q_2 = q_3 = q_4 = 1$, $q_5 = 2$. Here, the minimum number of vehicles required to satisfy the demand of S is $V(S) = 2$. Sharing the demand of customer 5 equally between the two vehicles yields the feasible solution shown in Fig. 1. However, constraint (12) applied to S is not satisfied since its left-hand side is equal to 4, while its right-hand side is equal to 3.

It is straightforward, however, to prove that constraints (13) are still valid for the SDVRP. This apparent contradiction is explained by the fact that the equivalence between (12) and (13) holds as long as the incoming (or outgoing) degree of vertex of S is equal to 1 in the optimal solution (see [12] for a discussion of this property in the case of undirected graphs). Clearly this is not the case in the SDVRP. However, a valid equivalence can still be derived. Let d_i denote the outgoing degree of vertex i:

$$d_i = \sum_{v=1}^{\bar{m}} \sum_{j=0}^{n} x_{ijv} \quad (i \in N). \tag{14}$$

We then prove the following equivalence.

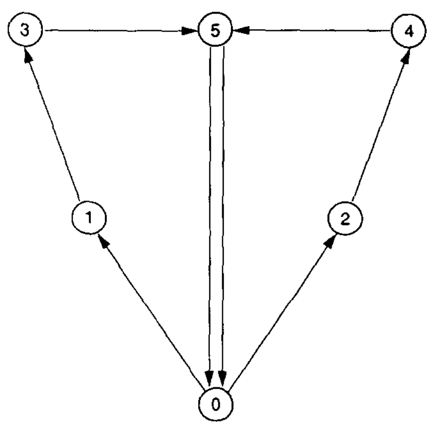

Fig. 1. Counter example showing that the classical VRP subtour elimination constraints (12) are invalid for the SDVRP.

Proposition 3.1. *The constraints*

$$\sum_{v=1}^{\bar{m}} \sum_{i,\, j \in S} x_{ijv} \leqslant \sum_{i \in S} d_i - V(S) \quad (S \subseteq N\backslash\{0\};\, |S| \geqslant 2) \tag{15}$$

are equivalent to constraints (13) *and are therefore valid inequalities for the SDVRP.*

Proof. The result follows immediately from the fact that for any nonempty subset S of $N\backslash\{0\}$, the following relationship is true by definition of d_i:

$$\sum_{i \in S} d_i = \left(\sum_{v=1}^{\bar{m}} \sum_{i,\, j \in S} x_{ijv} + \sum_{i \in S,\, j \in \bar{S}} x_{ijv} \right). \quad \square \tag{16}$$

In the SDVRP, constraints (15) can be imposed to eliminate subtours disconnected from the depot. Vehicle routes connected to the depot but having a total demand exceeding the vehicle capacity are prevented by (4).

Another concept closely related to subtours is that of *k-split cycles.*

Definition 3.2. Let $S = \{i_1, \ldots, i_k\} \subseteq N\backslash\{0\}$ and $k > 1$. If there exist h vehicle routes such that i_l and i_k are on the same route, and that for every $t = 1, \ldots, k-1$, i_t and i_{t+1} are on the same route, then S is k-split cycle.

The original Dror and Trudeau [8] definition of a k-split cycle was restricted to the case where $h = k$. This is, however, not necessary for our purpose and the required properties hold if the same route includes, for example, several pairs of vertices of S. If $h = 1$, the k-split cycle corresponds to a standard one-vehicle subtour. If $h = k$, then every customer of S (the k-split cycle) is split between exactly two vehicles. In what follows, it is implicitly assumed that $1 \leqslant h \leqslant k$. Dror and Trudeau [8] have proved the following result.

Proposition 3.3. *If* (c_{ij}) *satisfies the triangle inequality, there always exists an optimal SDVRP solution not containing k-split cycles.*

Note that if the triangle inequality is always strictly satisfied (i.e., $c_{ik} < c_{ij} + c_{jk}$ for all i, j, k), then no optimal SDVRP solution contains a k-split cycle. It is obvious from Definition 3.2 that if $S = \{i_1, \ldots, i_k\}$ defines a k-split cycle, then there exists a set S' satisfying $S \subseteq S' \subseteq N\backslash\{0\}$ and such that once arc directions are removed, the elements of S' form a cycle. This is illustrated in Fig. 2 (here $S' = \{1,2,3,4,5,6\}$). Similarly any (undirected) cycle defined on a subset S of $N\backslash\{0\}$ trivially defines an $|S|$-split cycle. It is therefore valid to eliminate all cycles on $N\backslash\{0\}$. We have thus proved the following result.

Proposition 3.4. *If* $C = (c_{ij})$ *satisfies the triangle inequality, the constraints*

$$\sum_{v=1}^{\bar{m}} \sum_{i,\, j \in S} x_{ijv} \leqslant |S| - 1 \quad (S \subseteq N\backslash\{0\};\, |S| \geqslant 2) \tag{17}$$

are valid inequalities for the SDVRP.

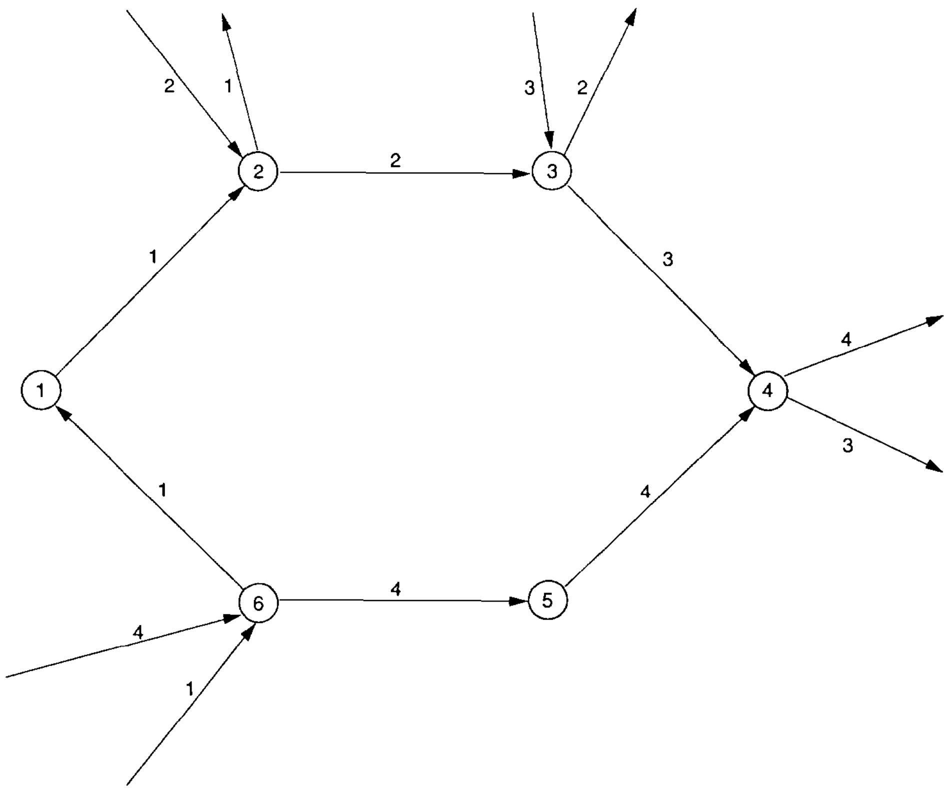

Fig. 2. $S = \{2, 3, 4, 6\}$ is a 4-split cycle. The values shown on the arcs are vehicle indices. When arc directions are removed, the element of S belong to the cycle formed by vertices 1, 2, 3, 4, 5 and 6.

In the VRP solution, the number of positive x_{ijv} variables is equal to $n + m$. The following proposition shows that this number can be larger in the case of SDVRPs.

Proposition 3.5. *There always exists an optimal SDVRP solution in which the number of positive x_{ijv} variables is at most equal to $n + 2m - 1$. (In the case of strict triangle inequality, the number of positive of positive variables is at most $n + 2m - 1$ in any optimal solution.)*

Proof. Every variable x_{ijv} with value 1 in the SDVRP solution corresponds to an arc in the solution graph. The maximal number of such arcs incident to the depot is $2m$. The number of arcs not incident to the depot cannot exceed $n - 1$ (the number of arcs in a spanning tree over $N\backslash\{0\}$) since otherwise there would exist a k-split cycle on $N\backslash\{0\}$. $\square$

It is interesting to illustrate by means of a diagram the hierarchy between the various subtour elimination constraints developed for the TSP, the VRP and the

SDVRP. The expressions shown in Fig. 3 are four possible right-hand sides for a subtour eliminating constraint whose left-hand side is $\sum_{v=1}^{\bar{m}} \sum_{i,j \in S} x_{ijv}$. An arrow pointing from one constraint to another indicates that the former is stronger than the latter. As shown above, constraints (12) are valid for the VRP, but too strong for the SDVRP. Constraints (17′) are valid for the SDVRP, but dominated by (15) and (17). In practice we use constraints (13) (equivalent to (15)) to eliminate subtours disconnected from the depot and situations in which the total demand of a set of S of customers exceeds the total capacity of all vehicles assigned to S. In problems where (c_{ij}) satisfies the triangle inequality, we use (17) to eliminate k-split cycles. When $\bar{m} = 1$, the constraints reduce to the original TSP subtour elimination constraints.

4. Additional classes of valid constraints

In addition to subtour elimination constraints and k-split cycle constraints, the following classes of constraints are also valid.

4.1. Outgoing degree of the depot

Since at least $\underline{m}$ vehicles are used in the solution, it is valid to impose

$$\sum_{j=1}^{n} x_{0jv} = 1 \quad (v = 1, \dots, \underline{m}) \tag{18}$$

(assuming $Q_1 \geqslant Q_2 \geqslant \cdots \geqslant Q_{\bar{m}}$).

Similarly, x_{0jv} will be equal to 1 for a least one index j if vehicle v is used to visit a customer, therefore

$$\sum_{j=1}^{n} x_{0jv} \geqslant \left(\sum_{i=1}^{n} q_i y_{iv} \right) \Big/ Q_{\underline{m}+1} (v = \underline{m} + 1, \dots, \bar{m}). \tag{19}$$

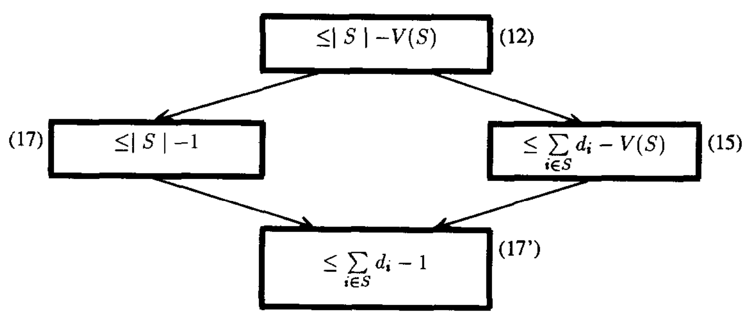

Fig. 3. Hierarchy between four types of subtour elimination constraints.

4.2. Variable fixing

When all vehicles have the same capacity, it is valid to assign one vehicle to one particular vertex i^*. We chose to assign vehicle 1 to the vertex i^* located the furthest away from the depot since this assignment has most impact on the lower bound:

$$\sum_{j=0}^{n} x_{i*j1} = 1. \tag{20}$$

In problems for which (c_{ij}) is symmetric, to avoid solutions which are merely symmetries of one another, it is valid to impose, for one arbitrary pair of cities $(\bar{i}, \bar{j})$, the constraint

$$\sum_{v=1}^{\bar{m}} x_{\bar{i}\bar{j}v} = 0. \tag{21}$$

4.3. Fractional cycle elimination constraints I

Consider any nonempty subset S of $N\setminus\{0\}$ and vehicle v. If v does not visit any vertex of S, then $\sum_{i,\,j\in S} x_{ijv} = \sum_{i\in S,\,j\in\bar{S}} x_{ijv} = 0$. If v visits at least one vertex of S then $\sum_{i,\,j\in S} x_{ijv} \leqslant |S| - 1$ and $\sum_{i\in S,\,j\in\bar{S}} x_{ijv} \geqslant 1$.

Thus, we have proved

Proposition 4.1. *The constraints*

$$\sum_{i\in S,\,j\in\bar{S}} x_{ijv} \geqslant \left(\sum_{i,\,j\in S} x_{ijv}\right)\bigg/(|S| - 1) \quad (S \subseteq N\setminus\{0\};\ |S| \geqslant 2;\ v = 1,\ldots,\bar{m}) \tag{22}$$

are valid inequalities for the SDVRP.

A graphical interpretation of constraints (22) is provided in Fig. 4. These constraints may sometimes be used to eliminate solution containing fractional cycles. To illustrate, consider the example shown in Fig. 5, with five nodes linked by four arcs traversed by the same vehicle. Suppose the values of the x_{ijv} variables are those shown on the arcs. Then constraints (22) are effective if S is defined as $\{1,2\}$ or as $\{3, 4\}$.

Figs. 6 and 7 illustrate what impact the introduction of a single constraint of type (22) can have on a solution. In Fig. 6, we have represented a fractional solution to a 20-vertex problem, satisfying constraints (2)–(5), (8), and $0 \leqslant x_{ijv} \leqslant 1$ for all i, j, v (in this solution, all x_{ijv} variables take the value $\frac{1}{2}$ unless otherwise indicated on the arcs). Fig. 7 is obtained by applying the same constraints, as well as constraint (22) for the $S = \{14, 19\}$. This results in a much simplified solution network, in which 2 vehicles are used instead of 5, and much closer to a feasible solution.

4.4. Fractional cycle elimination constraints II

The following proposition was suggested by Desrochers [6].

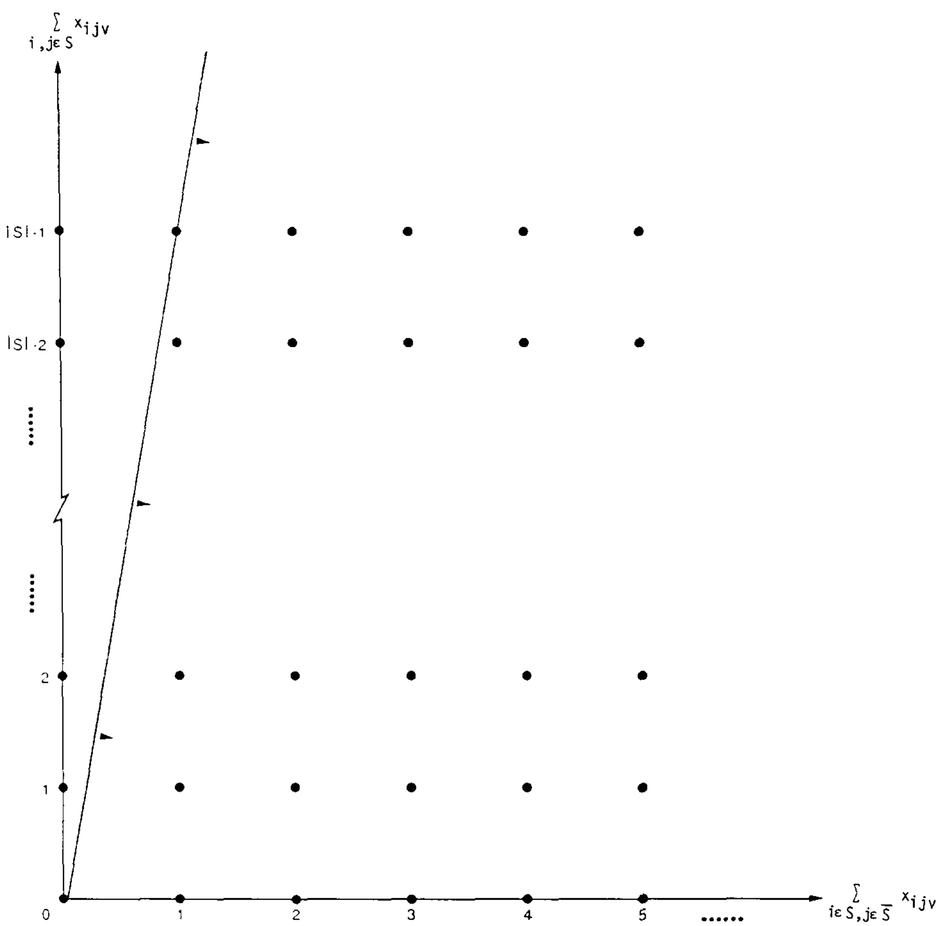

Fig. 4. Geometrical interpretation of constraints (22). Any feasible solution must lie below the line passing through (0, 0) and (1, $|S| - 1$).

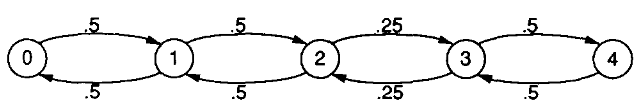

Fig. 5. Fractional cycles eliminated by constraints (22).

Proposition 4.2. *The constraints*

$$x_{ijv} \leqslant \sum_{k \neq i} x_{jkv} \quad (i, j \in N \setminus \{0\};\ v = 1, \ldots, \bar{m}) \tag{23}$$

are valid inequalities for the SDVRP.

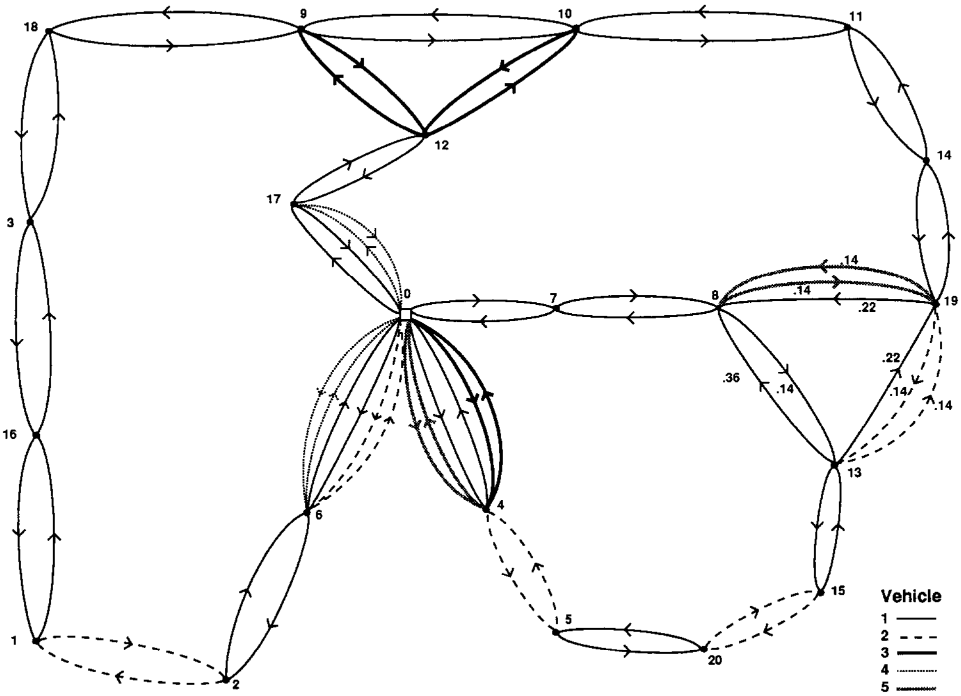

Fig. 6. Solution corresponding to the initial linear relaxation.

Proof. For any $i, j \in N\setminus\{0\}$ and for any v, constraints (2) imply

$$x_{ijv} \leqslant \sum_{k=0}^{n} x_{kjv} = \sum_{k=0}^{n} x_{jkv} = x_{jiv} + \sum_{k \neq i} x_{jkv}.$$

The conclusion follows immediately from the fact that $x_{jiv} = 0$ whenever $x_{ijv} = 1$. $\square$

There exist infeasible situations for which constraints (23) are violated while constraints (22) are satisfied. Consider, for example, the situation shown in Fig. 8. Constraints (22) are satisfied for all pairs $(i, i+1)$ $(i = 1, 2, 3)$ provided $\varepsilon > \frac{1}{4}$. However, constraint (23) is violated for $i = 3$, $j = 4$ since $x_{3,4,v} = 1 - 3\varepsilon$ and $\sum_{k \neq 3} x_{4,k,v} = 0 < 1 - 3\varepsilon$ when $\varepsilon < \frac{1}{3}$.

Another interesting relationship can be derived between constraints (22) and (23).

Proposition 4.3. *For $S = \{\bar{i}, \bar{j}\} \subseteq N\setminus\{0\}$, constraints* (22) *are implied by constraints* (23).

Proof. Summing up constraints (23) first defined for $i = \bar{i}$ and $j = \bar{j}$, and then for $i = \bar{j}$ and $j = \bar{i}$ yields $\sum_{k \neq \bar{i}} x_{\bar{j}kv} + \sum_{k \neq \bar{j}} x_{\bar{i}kv} \geqslant x_{\bar{i}\bar{j}v} + x_{\bar{j}\bar{i}v}$. This is precisely the expression of constraints (22) for $S = \{\bar{i}, \bar{j}\}$. $\square$

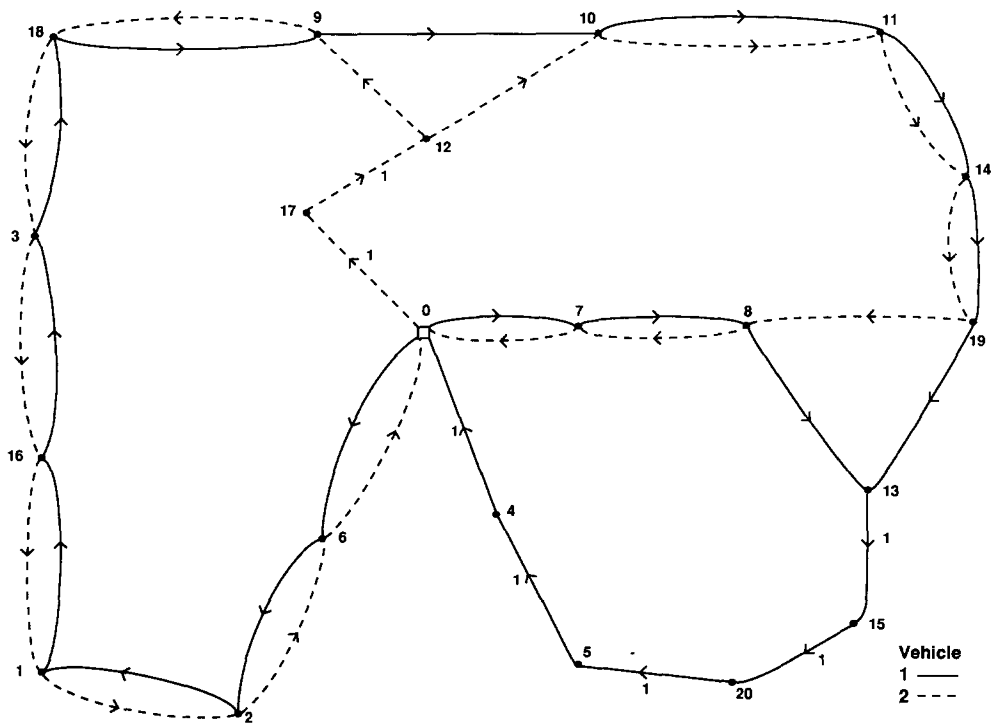

Fig. 7. Solution obtained imposing constraints (22) for $S = \{14, 19\}$.

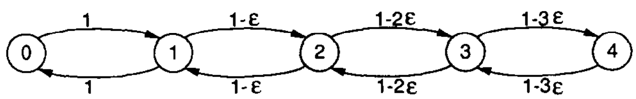

Fig. 8. Example for which constraints (23) are stronger than constraints (22).

Using this result, a hierarchy can now be established between constraints (22), (23), and those of Fig. 3. These relationships are valid for any subset S of $N \setminus \{0\}$, except for the comparison of (22) and (23) which has been derived only for the case $|S| = 2$. The hierarchy is is illustrated in Fig. 9.

5. Algorithm

We have developed the following constraint relaxation algorithm for the SDVRP.

Part 1: *Initialization*

Step 1: *Heuristic algorithm.* Obtain a first upper bound $\bar{z}$ on the value of the SDVRP solution by applying the heuristic algorithm developed by Dror and Trudeau [7].

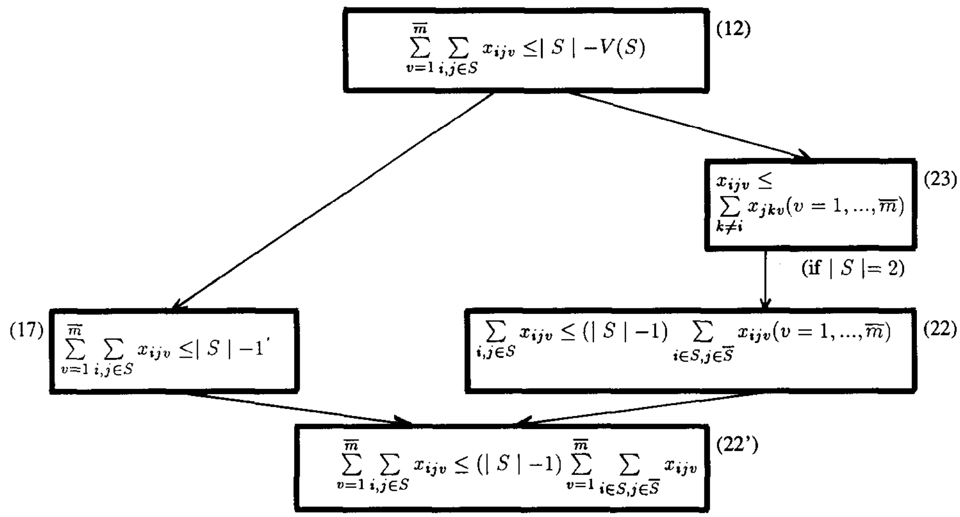

Fig. 9. Hierarchy between constraints (12), (17), (22) and (23).

Step 2: *Definition of a first subproblem.* Define a first subproblem consisting of (1)–(5), (8), the lower and upper bounds on the x_{ijv} variables, as well as a number of constraints of type (18)–(21) and (23).

Step 3: *First subproblem solution.* Solve the subproblem using simplex. If the solution is feasible for the SDVRP, the optimum has been reached: stop.

Step 4: *Checking for constraint violations.* Check for violations of constraints (13), (22) and (23). If no violated constraint can be identified, insert the subproblem in a stack and proceed to step 5. Otherwise, introduce a subset of all violated constraints which have been identified, and go to Step 3.

Part 2: *Branch and bound*

Step 5: *Termination check.* If the stack is empty, the optimum has been reached. Print the optimal solution, its value $\bar{z}$, and stop.

Step 6: *Subproblem selection.* Select a subproblem from the stack according to a "last in first out" criterion.

Step 7: *Subproblem solution.* Solve the subproblem using simplex and let $\underline{z}$ be its solution value. If $\underline{z} \geqslant \bar{z}$, go to Step 5. Otherwise, check whether the x variables are integer. If so, proceed to Step 9.

Step 8: *Subproblem partitioning.* The current subproblem solution is noninteger. Branch on a fractional x_{ijv} variable, thus creating two new subproblems which are then inserted in the stack. Go to Step 6.

Step 9: *Feasibility check.* Check whether the current integer solution contains subtours disconnected from the depot, k-split cycles, or vehicle routes whose total demand exceeds the vehicle capacity. Also check for violations of fractional cycle elimination constraints (23). If none can be identified, the solution is then feasible: set $\bar{z} := \underline{z}$, store the solution and proceed to Step 5. Otherwise, introduce a subset of all violated constraints (13), (17), (22) or (23) that have been identified, and go to Step 7.

In Steps 2, 4 and 9, several strategies are possible for the generation of violated constraints. (Note that we have chosen to detect violated k-split cycle constraints (17) only at an integer solution, in Step 9, in order to simplify the identification process.) In Step 2, we experimented with two policies. *Policy* 1: generate no constraint of type (18)–(21) or (23); *Policy* 2: generate all constraints of type (18)–(21) (since their number is only of the order of n) as well as constraints (23) for $v = 1, \dots, \underline{m}$ and all pairs (i, j) satisfying $c_{ij} \leqslant f$, where f is a control parameter. (The value if f is selected so as to avoid generating too many constraints. The value $\underline{m}$ is selected since the number of vehicles in the optimal solution is not yet known, but is certainly at least equal to $\underline{m}$). In Steps 4 and 9, all violated constraints (13) and (17) were generated; in addition, if constraint (23) was violated for some pair (i, j), then it was generated for that pair and for $v = 1, \dots, \bar{m}$.

Constraints (13), (17), (22) and (23) are generated as follows. For constraints (13), connected components that include the depot are first identified by means of a labelling procedure. Then S is successively defined as the set of all vertices included in each connected component, excluding the depot, and violations of (13) are then identified for each S in a straightforward manner. For constraint (17), all elementary circuits not including the depot and made up of arcs with positive x_{ijv}.

6. Computational results

In order to gain some insight into the efficiency of the linear relaxation, we carried out a series of computational experiments. We only analyse the behaviour of the algorithm at the root of the search tree (i.e., only Part 1 is executed), as this provides sufficient information on the tightness of the various cuts derived in this paper and we believe the computational burden of developing a full branch and bound scheme (with comparisons of branching criteria, variable fixing procedure, etc.) would add very little to the value of this study. Implementation details of various enumerative algorithms for similar routing problems are provided in the excellent survey by Balas and Toth [3].

All test problems were obtained by using a subset of n demand points of the 75-city problem described by Eilon et al. [9]. Note that in this problem, (c_{ij}) is symmetric and satisfies the triangle inequlity. (This type of routing problem is usually the hardest.) All vehicle capacities were set equal to the same constant $Q = 100$. For each problem, customer demands q_i were generated according to a uniform distribution in $[\alpha Q, \beta Q]$, where α and β are two control parameters. The value of $\underline{m}$ was set equal to $\lceil \sum_{i=}^{n} q_i / Q \rceil$, where $\lceil x \rceil$ denotes the smallest integer greater than or equal to x. The value of $\bar{m}$ was taken as n, since star-shaped solutions are always possible, but in practice this bound was never binding. All problems were solved solved on a SUN 3/50-4 work station.

Five problems were generated for each of the sizes $n = 10, 15$ and 20, with $[\alpha, \beta] = [0.1, 0.5]$. The following values of f were selected after limited experimentation: $f = 25$ for $n = 10$, and 15, $f = 10$ for $n = 20$. In each case, the value $\underline{z}$ of the lower bound provided by LP relaxation was computed after the initial problem solution, for each of the two constraint generation policies, and then, after completion of Part 1.

Table 1
Computational results for a fixed weight distribution, and various values of n

n	Problem number		After initial LP Policy 1	After initial LP Policy 2	At the end of Part 1
10	1	GAP (%)	35.66	9.46	5.18
		PIVOTS	249	322	710
		(13/17/23)		(–/–/38)	(5/1/38)
	2	GAP (%)	29.67	2.86	0.87
		PIVOTS	261	349	764
		(13/17/23)		(–/–/38)	(4/2/38)
	3	GAP (%)	37.76	5.75	1.32
		PIVOTS	208	403	1142
		(13/17/23)		(–/–/38)	(5/0/38)
	4	GAP (%)	33.78	8.47	6.01
		PIVOTS	273	313	1242
		(13/17/23)		(–/–/38)	(7/2/38)
	5	GAP (%)	29.15	3.07	0.00
		PIVOTS	216	316	1257
		(13/17/23)		(–/–/38)	(5/1/38)
15	1	GAP (%)	44.98	13.69	2.57
		PIVOTS	863	1360	9959
		(13/17/23)		(–/–/86)	(22/3/86)
	2	GAP (%)	43.33	17.21	8.38
		PIVOTS	840	1312	8642
		(13/17/23)		(–/–/86)	(25/2/86)
	3	GAP (%)	35.00	10.89	4.61
		PIVOTS	736	1516	6565
		(13/17/23)		(–/–/86)	(19/1/86)
	4	GAP (%)	33.75	10.42	7.15
		PIVOTS	813	1098	3687
		(13/17/23)		(–/–/86)	(10/2/86)
	5	GAP (%)	47.53	15.31	1.47
		PIVOTS	801	1586	18357
		(13/17/23)		(–/–/86)	(42/5/86)
20	1	GAP (%)	44.46	25.46	8.74
		PIVOTS	1387	2145	29112
		(13/17/23)		(–/–/16)	(39/9/176)
	2	GAP (%)	41.06	20.89	5.25
		PIVOTS	1575	1768	36817
		(13/17/23)		(–/–/16)	(42/8/116)
	3	GAP (%)	43.98	24.81	4.64
		PIVOTS	1468	1500	35454
		(13/17/23)		(–/–/16)	(47/10/176)
	4	GAP (%)	47.09	25.27	3.85
		PIVOTS	1612	2421	28709
		(13/17/23)		(–/–/16)	(43/11/196)
	5	GAP (%)	50.17	19.63	0.74
		PIVOTS	1602	1893	23727
		(13/17/23)		(–/–/16)	(37/6/176)

Table 2
Total pivot count for 5 problems, $n = 10$, and six weight distributions

Problem number	Weight distribution					
	[0.01, 0.1]	[0.1, 0.3]	[0.1, 0.5]	[0.1, 0.9]	[0.3, 0.7]	[0.7, 0.9]
1	804	855	710	3138	3190	8570
2	1236	1384	764	3961	3604	9488
3	1145	953	1142	5251	6816	2200
4	812	620	1242	4819	4331	2193
5	1200	561	1257	3496	4701	3118

Since the optimal value z^* was unknown, we used the ratio $(\bar{z} - \underline{z})/\bar{z}$ to measure the departure of $\underline{z}$ from the optimum. Here, $\bar{z}$ is the initial heuristic solution value. Note that this ratio overestimates $(z^* - \underline{z})/z^*$. The number of pivots and the number of constraints of each type that were generated are reported in Table 1. Aditional test were conducted for $n = 10, f = 25$ and various demand distribution parameters $[\alpha, \beta]$. For these problems, we only report the total number of pivots, which is a strong indicator of problem difficulty. The meanings of the various line headings in Table 1 are as follows: GAP(%): $100(\bar{z} - \underline{z})/\bar{z}$; PIVOTS: cumulative number of simplex pivots; (13/17/23): cumulative number of constraints of types (13), (17), and (23) that were generated.

The computational results suggest a number of observations. The various constraints developed for this problem were quite successful in reducing the gap between the lower and upper bounds at the root of the search tree. When constraints (13), (17) and (23) are used in conjunction, the value of the gap is contained between 0% and 9% on all test problems. However, the SDVRP seems considerably harder than the VRP (see, e.g., [16]) and it appears that branching will almost always be necessary. The low gaps obtained with our algorithm are a strong indication of the quality of the Dror and Trudeau heuristic. Also, our results are consistent with an observation made by these authors [8]: problems with small customer demands tend to require less computational effort (number of pivots) for their resolution. This is shown in Table 2.

7. Conclusion

We have considered in this paper a version of the vehicle routing problem in which split deliveries are allowed. Several families of valid inequalities were developed and hierarchy between these constraints was established. A constraint relaxation algorithm where branch and bound is used to achieve integrality was then described. It was shown how the various constraints developed help reduce the gap between the lower and upper bounds at the root of the search tree.

Acknowledgement

The authors are grateful to the Canadian Natural Sciences and Engineering Research Council for its financial support (grants OGP 0036429 and OGP 0039682). Thanks are also due to Martin Desrochers and to the referees for their helpful suggestions.

References

[1] K. Altinkemer and B. Gavish, Heuristics for unequal weight delivery problems with a fixed error guarantee, Oper. Res. Lett. 6 (1987) 149–158.
[2] K. Altinkemer and B. Gavish, Heuristics for delivery problems with constant error guarantees, Transportation Sci. 24 (1990) 294–297.
[3] E. Balas and P. Toth, Branch and bound methods, in: E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, eds., The Traveling Salesman Problem: A guided Tour of Combinatorial Optimization (Wiley, Chichester, UK, 1985) 361–401.
[4] N. Christofides, Vehicle routing, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, eds, The Traveling Salesman Problem: A guided Tour of Combinatorial Oprimization (Wiley, Chichester, UK 1985) 431–410.
[5] G.B. Dantzig, D.R. Fulkerson and S.M. Johnson, Solution of a large scale traveling-salesman problem, Oper. Res. 2 (1954) 393–410.
[6] M. Desrochers, Private communication (1989).
[7] M. Dror and P. Trudeau, Savings by split delivery routing, Transportation Sci. 23 (1989) 141–145.
[8] M. Dror and P. Trudeau, Split delivery routing, Naval Res. Logist. 37 (1990) 383–402.
[9] S. Eilon, C.D.T. Watson-Gandy, N. Christofides, Distribution Management: Mathematical Modelling and Practical Analysis (Griffin, London, 1971).
[10] B. Gavish, The delivery problem: new cutting planes procedures, paper presented at the TIMS XXVI Conference, Copenhagen (1984).
[11] B. Gavish, Augmented Lagrangean based algorithms for centralized network design, IEEE Trans. Comm. 33 (1985) 1247–1257.
[12] G. Laporte, Generalized subtour elimination constraints and connectivity constraints, J. Oper. Res. Soc. 37 (1986) 509–514.
[13] G. Laporte, The vehicle routing problem: an overview of exact and approximate algorithms, European J. Oper. Res. 59 (1992) 345–358.
[14] G. Laporte and Y. Nobert, A branch and bound algorithm for the capacitated vehicle routing problem, OR Spektrum 5 (1983) 77–85.
[15] G. Laporte and Y. Nobert, Exact algorithms for the vehicle routing problem, in: S. Martello, G. Laporte, M. Minoux and C. Ribeiro, eds., Surveys in Combinatorial Optimization, Annals of Discrete Mathematics, Vol. 31 (North-Holland, Amsterdam, 1987) 147–184.
[16] G. Laporte, Y. Nobert and M. Desrochers, Optimal routing under capacity and distance restrictions, Oper. Res. 33 (1985) 1050–1073.