Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

A Column Generation Approach to the Multiple-Depot Vehicle Scheduling Problem

Celso C. Ribeiro, François Soumis,

A COLUMN GENERATION APPROACH TO THE MULTIPLE-DEPOT VEHICLE SCHEDULING PROBLEM

CELSO C. RIBEIRO

Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil

FRANÇOIS SOUMIS

GERAD and École Polytechnique de Montréal, Montréal, Quebec, Canada

(Received July 1991; revision received December 1992; accepted February 1993)

We give a new formulation to the multiple-depot vehicle scheduling problem as a set partitioning problem with side constraints, whose continuous relaxation is amenable to be solved by column generation. We show that the continuous relaxation of the set partitioning formulation provides a much tighter lower bound than the additive bound procedure previously applied to this problem. We also establish that the additive bound technique cannot provide tighter bounds than those obtained by Lagrangian decomposition, in the framework in which it has been used so far. Computational results that illustrate the robustness of the combined set partitioning-column generation approach are reported for problems four to five times larger than the largest problems that have been exactly solved in the literature. Finally, we show that the gap associated with the additive bound based on the assignment and shortest path relaxations can be arbitrarily bad in the general case, and as bad as 50% in the symmetric case.

This paper addresses the multiple-depot vehicle scheduling problem (**MDVSP**), which consists of covering a set of predefined trips with vehicles provided by several depots. The problem appears, as an example, in urban bus scheduling, where, in most cases, the companies use many depots. The size of practical problems goes up to some hundreds of trips and ten depots. The problem is formulated in Carpaneto et al. (1989) as follows. A set of n trips $T_1, \ldots, T_n$ is given, trip T_j starting at time s_j and ending at time $e_j (j = 1, \ldots, n)$, as well as m depots $D_1, \ldots, D_m$ in the kth of which r_k vehicles are stationed $(k = 1, \ldots, m)$. Let τ_{ij} be the travel time for a vehicle to go from the ending point of trip T_i to the starting point of trip T_j. An ordered pair (T_i, T_j) is said to be compatible if and only if (iff) they can be covered by the same vehicle in the sequence, i.e., iff $e_i + \tau_{ij} \leqslant s_j$. Let c_{ij} be the finite cost incurred if a vehicle performs trip T_j immediately after trip T_i. For each trip T_j and each depot D_k, let $c_{n+k,j}$ (respectively, $c_{j,n+k}$) be the finite cost incurred if a vehicle stationed at depot D_k starts (respectively, ends) with trip T_j. The cost of a duty $(T_{i_1}, T_{i_2}, \ldots, T_{i_h})$ performed by a vehicle stationed at depot D_k is given by $c_{n+k,i_1} + c_{i_1,i_2} + \ldots + c_{i_{h-1},i_h} + c_{i_h,n+k}$. Then the problem consists of finding an assignment of trips to vehicles in such a way that each trip is covered by exactly one vehicle, each vehicle used in the solution covers a feasible duty (i.e., a sequence of pairwise compatible trips) and returns to its depot at the end of the duty, the number of vehicles leaving from depot D_k does not exceed $r_k (k = 1, \ldots, m)$, and the sum of the costs of the duties performed by the vehicles used in the solution is minimized.

The multiple-depot vehicle scheduling problem has been shown to be NP-hard when $m \geqslant 2$ (Bertossi, Carraresi and Gallo 1987). In the $m = 1$ case it is solvable in polynomial time as a minimum cost network flow problem. Notice that it can also be solved in polynomial time for $m \geqslant 2$ in the particular case in which the objective function corresponds to the minimization of the total number of vehicles used to perform the n trips. A heuristic algorithm based on Lagrangian relaxation is proposed in Bertossi,

Subject classifications: Transportation, mass transit: multiple-depot vehicle scheduling. Transportation, scheduling, vehicles: urban bus.
Area of review: Distribution, Transportation and Logistics.

003-365X/94/4201-0041$01.25

Carraresi and Gallo, and computational experiments are reported for problems with up to 50 trips and 3 depots. Another heuristic was proposed by Dell'Amico, Fischetti and Toth (1990). Their algorithm always guarantees the use of the minimum number of vehicles, and computational results are reported for larger problems with up to 500 trips and 4 depots. The errors are typically on the order of 1%. Other approximate, less specific algorithms are described in Bodin et al. (1978), Bodin, Rosenfield and Kydes (1978), Ceder and Stern (1981), Smith and Wren (1981), and El-Azm (1985). An exact branch-and-bound algorithm is given by Carpaneto et al. Their algorithm is based on the computation of lower bounds by an additive scheme. The gaps are relatively small and computational results are given for problems with up to 70 trips and 3 depots.

In this paper, we propose a new approach to the multiple-depot vehicle scheduling problem, based on the solution of its continuous relaxation by a column generation scheme. The paper is organized as follows. In Section 1 we establish that the additive bound technique (Carpaneto, Fischetti and Toth 1989, Fischetti and Toth 1988, 1989) cannot provide tighter bounds than those obtained by Lagrangian decomposition, in the framework in which it has been used so far. Section 2 gives the mathematical formulation of the multiple-depot vehicle scheduling problem as an integer program and we compare several lower bounds to the latter. We show that the bound given by the linear relaxation of the integer multicommodity flow formulation is at least as good as the additive bound proposed in Carpaneto et al. (1989), based on the assignment and shortest path relaxations. Following, Section 3 gives a new formulation of **MDVSP** as a set partitioning problem with side constraints, as well as a column generation scheme for the exact solution of its continuous relaxation. We show that the bound obtained by column generation is equal to that given by the linear relaxation of the integer multicommodity flow formulation, hence it is better than the additive bound. Computational results for problems with up to 300 trips and 6 depots are reported in Section 4. These results illustrate the robustness of the column generation approach, which obtains optimal solutions for much larger problems in smaller computational times, with respect to the branch-and-bound algorithm described in Carpaneto et al. The worst-case behavior of different bounds to the multiple-depot vehicle scheduling problem is evaluated in Section 5, in terms of the maximal gap with respect to the optimal solution. Finally, some conclusions are drawn in the last section.

1. ADDITIVE BOUND AND LAGRANGIAN DECOMPOSITION

For every optimization problem **P**, we denote by $v(\mathbf{P})$ its optimal value. Moreover, if **P** is an integer programming problem, we denote by $\overline{\mathbf{P}}$ its continuous relaxation, obtained by dropping all integrality constraints from the latter. We consider in this section the additive bound technique in the framework in which it is applied to an integer programming problem formulated as follows.

Problem $\mathbf{P}_{\mathcal{R}}$

Minimize $c \cdot x$

subject to $x \in \mathcal{R} = \mathcal{R}_1 \cap \mathcal{R}_2 \cap \ldots \cap \mathcal{R}_q$,

in which each set $\mathcal{R}_k$, $k = 1, \ldots, q$, is a polytope defined by linear inequalities and $\mathcal{R}_q$ also contains some integrality requirements. For a given cost vector c^k, let $\mathbf{P}_{\mathcal{R}_k}$ be the problem:

Problem $\mathbf{P}_{\mathcal{R}_k}$

Minimize $c^k \cdot x^k$

subject to $x^k \in \mathcal{R}_k$.

In many applications of the additive bound technique only the last problem $\mathbf{P}_{\mathcal{R}_q}$ is an integer programming problem, all others $\mathbf{P}_{\mathcal{R}_1}, \ldots, \mathbf{P}_{\mathcal{R}_{q-1}}$ are linear problems. The additive bound (Carpaneto, Fischetti and Toth 1989, Fischetti and Toth 1988, 1989) associated with the family of relaxations $\{\mathbf{P}_{\mathcal{R}_k}, k = 1, \ldots, q\}$ can be defined by

$$z_{\mathrm{ADD}} = \sum_{k=1}^{q} v(\mathbf{P}_{\mathcal{R}_k}),$$

where $c^1 = c$ and, for every $k = 1, \ldots, q-1$, $c^{k+1} \geqslant 0$ is a vector of residual costs satisfying $v(\mathbf{P}_{\mathcal{R}_k}) + c^{k+1} \cdot x \leqslant c^k \cdot x$, for all $x \in \mathcal{R}$. We show in the following that the additive bound z_{ADD} cannot be better than that obtained by Lagrangian decomposition.

Proposition 1. *Let* $\mathbf{P}_{\mathcal{R}_k}$ be a linear programming problem with reduced costs $\bar{c}^k$ associated with its optimal solution and take $c_j^{k+1} \in [0, \bar{c}_j^k]$ for $j = 1, \ldots, q$. Then the optimal solution to $\mathbf{P}_{\mathcal{R}_k}$ does not change if we take $c^k - c^{k+1}$ instead of c^k as its cost vector.

Proof. Let $\bar{x}^k$ be the optimal solution of $\mathbf{P}_{\mathcal{R}_k}$. Then $\bar{x}^k$ is also an optimal solution to the modified problem, because it is also feasible for the latter and the modified reduced costs are nonnegative.

We now give an alternative formulation of $\mathbf{P}_{\mathscr{R}}$, which is amenable to be dealt with by Lagrangian decomposition.

Problem $\mathbf{P}'_{\mathscr{R}}$

subject to $x^1 \in \mathscr{R}_1$

Minimize $c \cdot x^1$

$$x^1 = x^2, \quad x^2 \in \mathscr{R}_2$$
$$\ldots$$
$$x^{q-1} = x^q, \quad x^q \in \mathscr{R}_q.$$

Let $\lambda^2, \ldots, \lambda^q$ be dual multipliers associated with the equality constraints in the above formulation. then,

$$L(\lambda^2, \ldots, \lambda^q) = \text{minimize} \quad c \cdot x^1 + \sum_{k=2}^{q} \lambda^k \cdot (x^k - x^{k-1})$$
$$\text{subject to} \quad x^1 \in \mathscr{R}_1, \; x^2 \in \mathscr{R}_2, \; \ldots, \; x^q \in \mathscr{R}_q,$$

is the Lagrangian function associated with $\mathbf{P}'_{\mathscr{R}}$. The next result follows.

Proposition 2. *If* $\mathbf{P}_{\mathscr{R}_1}, \ldots, \mathbf{P}_{\mathscr{R}_{q-1}}$ are linear programming problems, then $z_{\text{ADD}} = L(c^2, \ldots, c^q)$.

Proof. We recall from Carpaneto et al. that the residual costs satisfy the inequality $0 \leqslant c^{k+1} \leqslant \bar{c}^k$. By definition,

From Proposition 1,

$$z_{\text{ADD}} = \sum_{k=1}^{k=q-1} \min\{(c^k - c^{k+1}) \cdot x^k \,|\, x^k \in \mathscr{R}_k\} + \min\{c^q \cdot x^q \,|\, x^q \in \mathscr{R}_q\}.$$

$$z_{\text{ADD}} = \sum_{k=1}^{k=q} \min\{c^k \cdot x^k \,|\, x^k \in \mathscr{R}_k\}.$$

Now, since the problems in the summation are independent,

$$z_{\text{ADD}} = \min\left\{ \sum_{k=1}^{k=q-1} (c^k - c^{k+1}) \cdot x^k + c^q \cdot x^q \,|\, x^1 \in \mathscr{R}_1, \ldots, x^q \in \mathscr{R}_q \right\}.$$

Finally, rearranging the terms in the latter expression, we obtain

$$z_{\text{ADD}} = \min\left\{ c^1 \cdot x^1 + \sum_{k=2}^{k=q} c^k \cdot (x^k - x^{k-1}) \,|\, x^1 \in \mathscr{R}_1, \ldots, x^q \in \mathscr{R}_q \right\} = L(c^2, \ldots, c^q).$$

As will be seen in Section 2, in the case of **MDVSP** problem $\mathbf{P}_{\mathscr{R}_q}$ has the integrality property. In that situation, the following result holds.

Theorem 1. *If* $\mathbf{P}_{\mathscr{R}_1}, \ldots, \mathbf{P}_{\mathscr{R}_{q-1}}$ *are linear programming problems and* $\mathbf{P}_{\mathscr{R}_q}$ *has the integrality property, then* $z_{\text{ADD}} \leqslant v(\mathbf{P}_{\mathscr{R}})$.

Proof. Here $z_{\text{ADD}} = L(c^2, \ldots, c^q) \leqslant \text{maximum}_{\lambda^2, \ldots, \lambda^q}\{L(\lambda^2, \ldots, \lambda^q)\} = v(\bar{\mathbf{P}}_{\mathscr{R}})$.

We notice that if the integrality property does not hold for $\mathbf{P}_{\mathscr{R}_q}$, then the additive bound z_{ADD} could eventually be better (i.e., larger) than the linear programming bound $v(\bar{\mathbf{P}}_{\mathscr{R}})$.

2. MODEL FORMULATION AND LOWER BOUNDS

We first give the mathematical formulation of the multiple-depot vehicle scheduling problem as an integer program. Let $N = \{1, \ldots, n\}$ represent the set of trips and $K = \{1, \ldots, m\}$ the set of depots. With each depot $k \in K$ we associate the graph $G^k = (V^k, A^k)$, where $n + k$ denotes the kth depot, $V^k = N \cup \{n + k\}$, and $A^k = N \times N \cup \{n + k\} \times N \cup N \times \{n + k\}$. Let x_{ij}^k be the flow of type k (i.e., the number of vehicles leaving from depot k) through arc $(i, j) \in A^k$. Then, the multiple-depot vehicle scheduling problem can be formulated as the following integer multicommodity flow problem.

Problem MDVSP

$$\text{Minimize} \quad \sum_{k=1}^{m} \sum_{(i,j)\in A^k} c_{ij} x_{ij}^k$$

$$\text{subject to} \quad \sum_{k=1}^{m} \sum_{i \in V^k} x_{ij}^k = 1 \quad \text{for all } j \in N \tag{1}$$

for all $k \in K$,

for all $j \in A^k$ (2)

$$\sum_{j \in N} x_{n+k,j}^k \leqslant r_k \quad \text{for all } k \in K$$

$$x_{ij}^k \geq 0 \quad \text{for all } k \in K, \text{ for all } (i,j) \in A^k \tag{3}$$

$$x_{ij}^k \text{ integer} \quad \text{for all } k \in K, \text{ for all } (i,j) \in A^k.$$

Constraints (1) ensure that each trip is performed exactly once, while (2) and (3) are, respectively, flow conservation and capacity constraints. Notice that x_{ij}^k takes binary values for all $i, j \in N$. We also remark that this is the appropriate formulation implicitly used in Carpaneto et al., although the authors gave another formulation with a weaker continuous relaxation.

We will now establish the relationship between the bounds obtained by the assignment relaxation, the shortest path relaxation, the additive technique, and linear programming. We show that the linear relaxation of **MDVSP** provides a stronger lower bound than does the additive bound of Carpaneto et al.

Assignment Bound. The assignment relaxation is obtained from $\overline{\mathbf{MDVSP}}$ by the aggregation of (2) into a surrogate constraint. If we define

$$x_{ij} = \sum_{k=1}^{k=m} x_{ij}^k, \quad V = \bigcup_{k=1}^{k=m} V^k = N \cup \left\{ \bigcup_{k=1}^{k=m} \{n+k\} \right\},$$

and $A = \cup_{k=1}^{k=m} A^k$, the relaxed problem is:

Problem AP

$$\text{Minimize} \sum_{(i,j)\in A} c_{ij} \cdot x_{ij} \tag{4}$$

$$\text{subject to} \quad \sum_{i\in V} x_{ij} = 1 \quad \text{for all } j \in N \tag{5}$$

$$\sum_{i\in V} x_{ij} - \sum_{i\in V} x_{ji} = 0 \quad \text{for all } j \in V \tag{6}$$

$$\sum_{j\in N} x_{n+k,j} \leq r_k \quad \text{for all } k \in K$$

$$x_{ij} \geq 0 \quad \text{for all } (i,j) \in A.$$

We will show that **AP** is equivalent to an assignment problem. Constraints (5) and (6) are equivalent to:

$$\sum_{i\in V} x_{ij} = 1 \quad \text{for all } j \in N \tag{5}$$

$$\sum_{j\in V} x_{ij} = 1 \quad \text{for all } i \in N \tag{7}$$

$$\sum_{i\in V} x_{ij} - \sum_{i\in V} x_{ji} = 0 \quad \text{for all } j \in V\backslash N. \tag{8}$$

If we define $x_{n+k,n+k}$ as the number of vehicles that do not leave from depot D_k among the r_k stationed there, (8) can be written for all $j \in V\backslash N$ as:

$$\sum_{i\in V} x_{i,n+k} = r_k \quad \text{for all } k \in K \tag{9}$$

$$\sum_{j\in V} x_{n+k,j} = r_k \quad \text{for all } k \in K. \tag{10}$$

Problem **AP** is then equivalent to the transportation problem (4), (5), (7), (9) and (10). The latter can be written as an assignment problem by the use of r_k copies of each depot, each representing one among the r_k vehicles stationed. Hence, the bound $v(\mathbf{AP})$ can be obtained by the solution of an assignment problem.

Shortest Path Bound. The path relaxation associated with node $v \in N$ is obtained from **MDVSP** by dropping (1) for all $j \neq v, j \in N$:

Problem SP$_v$

$$\text{Minimize} \sum_{k=1}^{m} \sum_{(i,j)\in A^k} c_{ij} x_{ij}^k$$

subject to

$$\sum_{k=1}^{m} \sum_{i\in V^k} x_{iv}^k = 1$$

$$\sum_{i\in V^k} x_{ij}^k - \sum_{i\in V^k} x_{ji}^k = 0 \quad \text{for all } k \in K \quad \text{for all } j \in V^k$$

$$x_{ij}^k \geq 0 \quad \text{for all } k \in K \quad \text{for all } (i,j) \in A^k.$$

A feasible solution to $\mathbf{SP}_v$ is a circulation of one unity of flow passing through node v. As all circuits need to pass through some depot, the bound $v(\mathbf{SP}_v)$ is equal to the cost of the shortest circuit passing through nodes v and $n + k$ for some $k \in K$. The cost of this circuit is given by the sum of the costs of the shortest paths from node $n + k$ to node v and from node v to node $n + k$.

Additive Bound. The additive bound for **MDVSP** is computed as follows (see Carpaneto et al. for the details). First, set the additive bound z_{ADD} equal to $v(\mathbf{AP})$, the optimal value of the assignment relaxation. Then, for each depot D_k and for each trip T_j, compute $\delta_{kj} = \lambda_{kj} + \overline{\lambda}_{kj}$, where λ_{kj} (respectively, $\overline{\lambda}_{kj}$) is the shortest path from depot D_k (respectively, the node associated with trip T_j) to the node associated with trip T_j (respectively, depot D_k) without visiting any other depot, in terms of the reduced costs defined by the dual solution of **AP**. Next, set $z_{ADD} \leftarrow z_{ADD} + \delta_{pl}$, where $\delta_{pl} = \text{maximum}_{k=1,\dots,m;j=1,\dots,n}\{\delta_{kj}\}$. The dual solution of the shortest path problem from depot D_p to trip T_l defines new residual costs. A new iteration resumes by the computation of a new shortest path bound in terms of such residual costs, until no further improvement can be attained. In this scheme, the assignment bound exploits the condition that each trip must be covered exactly once, while the shortest path bounds capture the fact that each vehicle performing a duty should return to the same depot from where it left. From the results in this section, we have the following theorem.

Theorem 2

i. $\text{Maximum}_{v\in N}\{v(\mathbf{SP}_v)\} \leq v(\overline{\mathbf{MDVSP}})$.
ii. $v(\mathbf{AP}) \leq z_{add}$.
iii. $z_{ADD} \leq v(\overline{\mathbf{MDVSP}})$.

Proof

i. Trivial, because $\mathbf{SP}_v$ for all $v \in N$, is a relaxation of $\overline{\mathbf{MDVSP}}$.
ii. Trivial, from the definition of the additive bound.
iii. The residual costs defined in Carpaneto et al. pp. 536–539) satisfy the conditions of Proposition 1. We now proceed with the reformulation of **MDVSP**, as in Section 1 for the integer programming problem $\mathbf{P}_{\mathscr{R}}$. Take $\mathscr{R}_1$ as the set of constraints associated with the assignment relaxation. Next, for every $k = 2, \dots, n + 1$, take $\mathscr{R}_k$ as the set of constraints associated with the shortest path relaxation $\mathbf{SP}_v$, for some $v \in N$, in the same order in which the latter would be solved during the computation of the additive bound. Give different names to the variables appearing in each block of constraints and use equality constraints to make them be the same. This reformulation of **MDVSP** is amenable to be dealt with by Lagrangian decomposition. The result follows from Theorem 1.

3. A COLUMN GENERATION APPROACH

MDVSP can be reformulated in terms of variables associated with the circuits of the graphs $G^k = (V^k, A^k)$ defined in the Introduction. For every $k \in K$, let Ω_k be the set of paths leaving from depot D_k, visiting some nodes (trips) of N, and coming back to the same depot. For every path $p \in \Omega = \cup_{k=1}^{k=m} \Omega_k$, let c_p be the sum of the costs of its arcs and $a_{jp} = 1$ iff it visits node $j \in N$, $a_{jp} = 0$ otherwise. Now, if we associate a 0–1 variable y_p with every path $p \in \Omega$, we obtain the equivalent formulation:

Problem MDVSP′

$$\text{Minimize} \quad \sum_{k=1}^{m} \sum_{p\in\Omega_k} c_p y_p$$

$$\text{subject to} \quad \sum_{k=1}^{m} \sum_{p\in\Omega_k} a_{jp} y_p = 1 \quad \text{for all } j \in N \qquad (11)$$

$$\sum_{p\in\Omega_k} y_p \leq r_k \quad \text{for all } k \in K \qquad (12)$$

$$y_p \in \{0, 1\} \quad \text{for all } p \in \Omega.$$

The set partitioning constraints (11) ensure that each node $j \in N$ has to be visited by exactly one circuit (i.e., each trip has to be performed by one duty). The cardinality constraints (12) ensure that no more than r_k vehicles stationed at the depot D_k, $k \in K$ will be used. The continuous relaxation $\overline{\mathbf{MDVSP'}}$ can be solved by column generation. Let π_j for all $j \in N$, and σ_k for all $k \in K$, be the dual variables associated with (11) and (12), respectively. The subproblem corresponding to the generation of the column with minimum marginal cost among those of Ω_k consists of finding a shortest path leaving from depot D_k, visiting at least one node of N, and coming back to the same depot.

Problem $\mathbf{CG_k}$

$$\text{Minimize} \quad -\sigma_k + \sum_{(i,j)\in A^k} (c_{ij} - \pi_j)x_{ij}^k$$

$$\text{subject to} \quad \sum_{i\in V^k} x_{ij}^k - \sum_{i\in V^k} x_{ji}^k = 0 \quad \text{for all } j \in N$$

$$\sum_{i\in N} x_{n+k,i}^k = 1$$

$$\sum_{i\in N} x_{i,n+k}^k = 1$$

$$x_{ij}^k \in \{0, 1\} \quad \text{for all } (i, j) \in A^k.$$

Let $\bar{x}_{ij}^k$ for all $(i, j) \in A^k$ be the optimal solution of $\mathbf{CG_k}$. Then, a new column p to be added to $\overline{\mathbf{MDVSP'}}$ is obtained as: Take $c_p = \sum_{(i,j)\in A^k} c_{ij}\bar{x}_{ij}^k$ and $a_{jp} = \sum_{i\in V^k} \bar{x}_{ij}^k$ for all $j \in N$. From Theorem 2 and the results in this section, we can now prove the following theorem concerning the column generation bound.

Theorem 3. $v(\overline{\mathbf{MDVSP}}) = v(\overline{\mathbf{MDVSP'}})$.

Proof. It is clear that **MDVSP** and **MDVSP′** are equivalent formulations of the same 0–1 problem. We now show that their linear relaxations are also equivalent. Suppose that we are solving $\overline{\mathbf{MDVSP}}$ by Dantzig-Wolfe decomposition (see e.g., Lasdon 1970) with constraints (1) and (3) in the master problem. For each $k \in K$, the subproblem defined by constraints

(2) and the nonnegativity constraints corresponds to finding a minimal cost circulation through depot D_k, in terms of the reduced costs defined by the dual variables of the master problem. The solution of this subproblem is either zero (when all circuits have a positive reduced cost and optimality was attained) or unbounded. In the latter case, it is a circuit through depot D_k, associated with an extreme ray of the polytope defined by the constraints of the subproblem. Now, rewrite $\overline{\mathbf{MDVSP}}$ replacing each variable x_{ij}^k by a linear combination of the extreme rays of the polytopes associated with each subproblem. Since the solution of the subproblem is always an extreme ray, there is no convexity constraint. Each extreme ray associated with a depot D_k in the Dantzig-Wolfe formulation is the same as a path in Ω_k in $\overline{\mathbf{MDVSP}}$. It follows that the values of the coefficients of the current linear combination of extreme rays at each iteration of the $\overline{\mathbf{MDVSP}}$ and $\overline{\mathbf{MDVSP'}}$ are equivalent (see also Desrosiers, Soumis and Desrochers 1988 for a similar result for the minimum fleet size multiple traveling salesman problem with time windows).

Theorems 2iii and 3 show that the lower bound obtained by column generation is at least as good as that obtained by the additive technique, i.e., $v(\overline{\mathbf{MDVSP'}}) \geqslant z_{\mathrm{ADD}}$. This fact is in the origin of the excellent computational results presented and discussed in the next section.

4. COMPUTATIONAL RESULTS

The column generation approach to the solution of **MDVSP′** was implemented through the code GENCOL, which is a general purpose software for the solution of routing and scheduling problems formulated as set covering or set partitioning problems with side constraints (Sansó et al. 1990). GENCOL solves by column generation a master problem defined by the continuous relaxation of a set covering or set partitioning problem with side constraints. The column generation subproblems are solved by specialized shortest path algorithms. The embedded branch-and-bound algorithm supports different branching criteria and searching strategies.

In the case of the **MDVSP**, the column generation subproblem is an unconstrained shortest path problem in an acyclic graph (obtained by duplication of the depot nodes). CPLEX is used as the linear programming solver. Depth-first search is used as the branching strategy. All computational results reported below were obtained on a Sun Sparc 2 workstation.

Test problems were randomly generated as in Carpaneto et al., to simulate a real-life public transport system in which there are short and long trips running from 5 a.m. to midnight. Long trips correspond to extra-urban journeys, or to sequences of urban journeys. Short trips correspond to urban journeys with peak hours around 7–8 a.m. and 5–6 p.m. Let $\rho_1, \ldots, \rho_\nu$ be the set of relief points representing the points where trips can start or finish, randomly generated according to a uniform distribution in a 60×60 square. For each pair (a, b) of relief points, the travel time from a to b is given by the Euclidean distance between them. For each trip T_j, $j = 1, \ldots, n$, the starting and ending relief points, ρ_j^s and ρ_j^e, are generated as uniformly distributed random integers in $[1, \nu]$. In each pair of trips, T_i and T_j, let $\tau\rho_i^e\rho_j^s$ be the travel time from the ending point of trip T_i to the starting point of trip T_j. The starting and ending times of trip T_j were generated taking into account two classes of trips:

1. Short trips, with a probability of 40%: The starting time s_j of trip T_j is generated as a uniformly distributed random integer in [420, 479] with a probability of 15%, in [480, 1,019] with a probability of 70%, and in [1,020, 1,080] with a probability of 15% (these time intervals represent a journey of 11 hours, with two peak periods with the duration of one hour each). The ending time e_j of trip T_j is a random integer uniformly distributed in $[s_j + \tau\rho_j^s\rho_j^e + 5,\ s_j + \tau\rho_j^s\rho_j^e + 40]$.
2. Long trips, with a probability of 60%: Here s_j and e_j are random integers uniformly distributed, respectively, in [300, 1,200] and $[s_j + 180, s_j + 300]$. Moreover, every long trip T_j is circular, i.e., $\rho_j^e = \rho_j^s$.

Two classes of problems were considered. For class A, all the m depots were randomly located inside the 60×60 square. For class B and $m = 2, 3$, two depots were located in opposite corners of the square ($m = 2$) and the third one (in the $m = 3$ case) was randomly generated at a point inside the 60×60 square. The number r_k of vehicles available at each depot D_k was generated as an integer uniformly distributed in $[3 + n/(3m), 3 + n/(2m)]$. The costs are given by:

1. $c_{ij} = \lfloor 10\tau_{ij} + 2(s_j - e_i - \tau_{ij}) \rfloor$ for all compatible pairs (T_i, T_j);
2. $c_{n+k,j} = \lfloor 10 \cdot ED(D_k, \rho_j^s) \rfloor + 5{,}000$ for all depots D_k and trips T_j; and
3. $c_{j,n+k} = \lfloor 10 \cdot ED(\rho_j^e, D_k) \rfloor + 5{,}000$ for all depots D_k and trips T_j,

where $ED(a, b)$ denotes the Euclidean distance between a and b. This cost structure is such that the strongest costs are associated with the arcs leaving or entering the depots. Hence, the model finds the best solution (in terms of a mixture of travel and idle times) with the minimum number of vehicles which ensure feasibility.

In the basic set of test problems, eight values of n = 30, 40, . . . , 100 were considered for each value of m and for each class of test problems. The value of ν is a random integer uniformly distributed in $[n/3, n/2]$. For each class and for each value of m and n, ten instances of the **MDVSP** were generated. For each value of m and n, the following average results (over ten test problems) are given in Table I (class A) and Table II (class B):

Avgtime	average computation time;
Maxtime	maximum computation time;
Avgnodes	average number of nodes in the branch-and-bound tree;
Maxnodes	maximum number of nodes in the branch-and-bound tree;
Avggap	average integrality gap;
Maxgap	maximum integrality gap;
Avgcol	average number of columns generated;
Maxcol	maximum number of columns generated.

The continuous relaxation of **MDVSP′**, solved by column generation, provides a much better lower bound than the additive bounding scheme described in Carpaneto et al. The average integrality gap over all test problems in Tables I and II is only 0.00076%, while the average gap associated with the additive lower bound is on the order of 0.9%. While the average overall computational times to obtain the optimal integer solution grow very fast in Carpaneto et al. (e.g., from 3.5 seconds for n = 30, to 1756.0 seconds for n = 70 for class A and m = 2), they behave very well in the case of the column generation approach (ranging from 1.1 seconds for n = 30 to only 49.2 seconds for n = 100 on a slower machine; see Table I). All test problems in this first set were solved to optimality. The behavior of the algorithm does not seem to be affected by the class of the test problems.

To further investigate the behavior of the column generation approach, we fixed n = 100 and we solved larger problems in terms of the number of depots (m = 2, . . . , 10). The depots are located as follows: m = 2, two depots in opposite corners of the square; m = 3, the third depot is randomly generated inside the square; m = 4, four depots in the corners of the square; m = 5–10, four depots in the corners and the other randomly generated inside the square. The average numerical results (over ten problems for each value of m) are given in Table III, where it can be noticed that the computational times do not seem to explode with the increase in the number of depots.

We also solved much larger instances with up to 300 trips. For each value of m = 2, . . . , 6, we generated ten test problems for n = 150, 200, 250, 300. This final set of test problems is formed by instances four to five times larger than the largest problems exactly solved so far. The computational results are given in Table IV. The last column of this table shows the

Table I

Computational Results for Basic Test Problems for Class A

n	Avgtime (Seconds)	Maxtime (Seconds)	Avgnodes	Maxnodes	Avggap ($\times 10^{-4}$%)	Maxgap ($\times 10^{-3}$%)	Avgcol	Maxcol
m = 2 Depots								
30	1.1	2	1.0	1	0.0	0.0	149.5	163
40	2.4	5	1.4	5	2.4	2.4	219.9	294
50	3.3	4	1.0	1	0.0	0.0	274.8	295
60	7.7	14	1.5	3	1.7	0.9	369.8	445
70	13.0	18	1.2	2	0.0	0.0	471.3	531
80	16.5	30	1.4	5	4.5	4.5	533.8	663
90	26.5	69	2.8	19	5.4	5.4	637.0	813
100	49.2	190	8.7	73	3.9	2.7	765.5	897
m = 3 Depots								
30	1.4	2	1.0	1	0.0	0.0	171.9	198
40	2.3	3	1.2	3	2.7	2.7	241.6	271
50	5.7	22	2.8	19	20.0	20.0	327.1	468
60	7.0	10	1.0	1	0.0	0.0	406.0	466
70	12.0	19	1.9	5	15.0	9.1	499.9	599
80	18.0	46	2.2	13	2.2	2.2	569.3	723
90	29.4	49	2.2	7	13.0	7.2	714.6	885
100	32.8	42	1.4	3	0.2	0.2	780.3	852

Table II
Computational Results for Basic Test Problems for Class B

n	Avgtime (Seconds)	Maxtime (Seconds)	Avgnodes	Maxnodes	Avggap ($\times 10^{-4}$%)	Maxgap ($\times 10^{-3}$%)	Avgcol	Maxcol
m = 2 Depots								
30	1.2	2	1.2	3	0.0	0.0	151.4	169
40	3.0	6	3.0	15	21.0	17.0	231.3	307
50	4.3	6	1.4	3	1.5	1.5	288.7	335
60	5.9	7	1.0	1	0.0	0.0	354.1	376
70	12.3	18	1.3	3	1.2	1.2	462.2	530
80	16.4	26	1.4	4	3.5	3.5	521.5	605
90	23.9	63	3.0	21	4.6	4.6	595.1	760
100	33.0	48	1.6	7	3.2	3.2	712.9	806
m = 3 Depots								
30	1.5	2	1.3	3	6.4	6.4	179.6	203
40	3.0	4	1.0	1	0.0	0.0	249.3	272
50	6.3	22	4.0	23	24.0	20.0	343.5	541
60	10.7	36	3.9	21	29.0	19.0	429.5	606
70	17.3	70	5.2	39	2.2	1.4	511.4	694
80	19.2	29	2.0	9	3.1	1.6	591.6	712
90	30.5	41	2.2	8	5.3	3.6	717.2	792
100	45.1	92	4.8	19	13.0	6.8	830.5	978

Table III
Computational Results (Sensitivity to the Number of Depots, n = 100)

	Class B, n = 100							
m	Avgtime (Seconds)	Maxtime (Seconds)	Avgnodes	Maxnodes	Avggap ($\times 10^{-4}$%)	Maxgap ($\times 10^{-3}$%)	Avgcol	Maxcol
2	33.0	48	1.6	7	3.2	3.2	712.9	806
3	45.1	92	4.8	19	13.0	6.8	830.5	978
4	46.7	134	4.4	21	13.0	5.2	843.8	1,117
5	94.4	389	17.3	97	32.0	9.9	941.2	1,403
6	109.5	534	17.6	109	34.0	14.0	1,038.8	1,676
7	48.0	77	3.2	8	16.0	5.5	963.4	1,097
8	116.6	320	21.9	93	45.0	14.0	1,116.2	1,337
9	82.8	385	10.2	70	16.0	13.0	1,067.5	1,557
10	255.7	755	51.6	204	63.0	17.0	1,259.9	1,693

number of problems solved to optimality within the limit of 300 nodes in the branch-and-bound tree (200 nodes for the largest problems, with n = 250 and n = 300, due to the increase in the computational time), out of the ten test problems generated for each pair of values of m and n. Most of the test problems with n up to 300 were solved to optimality in reasonable computational times (averages are taken only over the problems solved to optimality within those limits on the number of nodes in the branch-and-bound tree).

The computational results presented in this section show that the column generation approach is very robust: The computational times are not affected by the class of the test problems and do not seem to explode with the increase in the number of depots. Moreover, the lower bounds do not deteriorate with the increase in problem size. The quality of the lower bounds made it possible to solve to optimality problems four to five times larger than the largest problems exactly solved in the literature so far.

5. WORST-CASE ANALYSIS

In this section we consider the worst-case behavior of lower bounds, with the respect to the cost of the optimal integer solution. This is an interesting type of analysis, which has not received too much attention. In the case of the **MDVSP**, we study the assignment bound, the shortest path bound, and the additive bound.

Theorem 4. *The worst-case ratio* $z_{\mathrm{ADD}}/v(\mathbf{MDVSP})$ *can be arbitrarily bad.*

Proof. We consider a family $\{\Gamma^1(n)\}$ of instances of **MDVSP**, each with two depots D_1 and D_2. For each

Table IV
Computational Results for Larger Test Problems

m	n	Avgtime (Seconds)	Maxtime (Seconds)	Avgnodes	Maxnodes	Avggap ($\times 10^{-4}$%)	Maxgap ($\times 10^{-3}$%)	Avgcol	Maxcol	Solved to Optimality
2	100	33.0	48	1.6	7	3.2	3.2	712.9	806	10/10
	150	268.7	1,152	23.0	151	8.2	2.6	1,341.2	1,783	10/10
	200	679.8	2,620	21.1	112	9.8	3.3	1,928.2	2,386	10/10
	250	1,254.2	4,764	9.4	51	3.8	1.6	2,687.5	3,541	10/10
	300	1,499.8	4,915	4.6	27	2.0	0.3	3,284.6	3,986	8/10
3	100	45.1	92	4.8	19	13.0	6.8	830.5	978	10/10
	150	254.2	544	16.7	51	24.0	5.7	1,441.2	1,623	10/10
	200	1,105.2	2,909	43.4	133	14.0	4.3	2,239.3	2,673	9/10
	250	1,704.6	3,838	20.4	67	10.0	2.5	2,883.9	3,399	7/10
	300	1,832.5	3,658	5.5	15	3.8	1.2	3,568.8	3,699	4/10
4	100	46.7	134	4.4	21	13.0	5.2	843.8	1,117	10/10
	150	655.2	2,962	55.3	261	29.0	7.5	1,664.2	2,437	10/10
	200	1,158.8	4,369	41.7	185	24.0	5.8	2,206.6	2,851	9/10
	250	6,230.0	11,179	91.3	160	41.0	5.2	3,600.7	4,329	3/10
	300	5,493.0	5,493	54.0	54	11.0	1.1	3,895.0	3,895	1/10
5	100	94.4	389	17.3	97	32.0	9.9	941.2	1,403	10/10
	150	939.6	5,889	92.1	614	17.0	6.0	1,738.3	3,182	10/10
	200	672.6	1,763	19.4	74	24.0	11.0	2,247.8	2,582	8/10
	250	1,920.7	3,634	33.5	79	11.0	2.1	3,133.0	3,324	6/10
	300	3,002.5	9,246	36.0	138	5.3	2.0	3,843.5	4,585	4/10
6	100	109.5	534	17.6	109	34.0	14.0	1,038.8	1,676	10/10
	150	682.7	2,500	66.1	298	36.0	8.6	1,803.3	2,315	10/10
	200	1,641.1	7,524	58.4	284	25.0	5.9	2,442.1	3,412	8/10
	250	4,562.5	9,038	88.3	179	18.0	2.8	3,568.2	4,123	6/10
	300	3,346.8	5,253	24.5	44	10.0	2.4	4,092.8	4,383	4/10

$k = 1, 2$, there are n trips $T_{j+(k-1)n}$, $j = 1, \ldots, n$ in a row. Travel times are taken as:

- 0, from each depot D_k to the starting point of trip $T_{1+(k-1)n}$;
- $n + 1$, from each depot D_k to the starting point of each trip $T_{j+(k-1)n}$, $j = 2, \ldots, n$;
- j, from the ending point of each trip $T_{j+(k-1)n}$, $j = 1, \ldots, n$ to depot D_k;
- 1, from each depot D_{k+1}(D_1, if $k = 2$) to the starting point of trip $T_{n+(k-1)n}$;
- $n + 1$, from each depot D_{k+1}(D_1, if $k = 2$) to the starting point of each trip $T_{j+(k-1)n}$, $j = 1, \ldots, n-1$;
- $n - j$, from the ending point of each trip $T_{j+(k-1)n}$, $j = 1, \ldots, n$ to depot D_{k+1}(D_1, if $k = 2$);
- 0, from the ending point of trip $T_{i+(k-1)n}$ to the starting point of trip $T_{j+(k-1)n}$ for all $1 \leq i < j \leq n$.

Moreover, we suppose that all other costs (travel times) are sufficiently large. The starting and ending times of each trip $T_{j+(k-1)n}$, $k = 1, 2, j = 1, \ldots, n$ are $s_{j+(k-1)n} = e_{j+(k-1)n} = j$. An optimal solution can be built by taking one vehicle from each depot D_k, $k = 1, 2$, and making it perform trips $T_{1+(k-1)n}, T_{2+(k-1)n}, \ldots, T_{n+(k-1)n}$ in this order. Then, $v(\mathbf{MDVSP}) = 2 \cdot n$.

Since $r_1 = r_2 = 1$ for the instances of family $\{\Gamma^1(n)\}$, the assignment relaxation **AP** can be rewritten in that case as:

$$\text{Minimum} \quad \sum_{(i,j)\in A} c_{ij} \cdot x_{ij}$$

$$\text{subject to} \quad \sum_{j\in V} x_{ij} = 1 \qquad \text{for all } i \in V \qquad (13)$$

$$\sum_{i\in V} x_{ij} = 1 \qquad \text{for all } j \in V \qquad (14)$$

$$x_{ij} \geq 0 \qquad \text{for all } (i, j) \in A,$$

with $V = \{D_1, T_1, T_2, \ldots, T_n, D_2, T_{n+1}, T_{n+2}, \ldots, T_{2n}\}$. The optimal solution $\bar{x}$ to the assignment relaxation is a single cycle leaving from and coming back to depot D_1, after visiting, in this order, the nodes associated with trips $T_1, T_2, \ldots, T_n$, depot D_2, and the nodes associated with trips $T_{n+1}, T_{n+2}, \ldots, T_{2n}$ (see Figure 1). Then, $v(\mathbf{AP}) = 0$ and

$$\bar{x}_{D_1T_1} = \bar{x}_{T_1T_2} = \ldots = \bar{x}_{T_nD_2} = \bar{x}_{D_2T_{n+1}}$$
$$= \bar{x}_{T_{n+1}T_{n+2}} = \ldots = \bar{x}_{T_{2n}D_1} = 1;$$

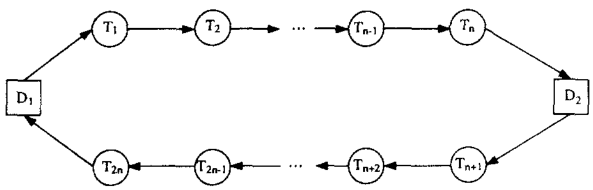

Figure 1. Optimal solution to the assignment relaxation of instance $\Gamma^1(n)$.

all other variables are equal to zero. Now, take the following values for the dual variables u_i for all $i \in V$, and v_j for all $j \in V$, associated with (13) and (14), respectively:

$$\bar{u}_{D_1} = \bar{v}_{D_1} = 0;$$

$$\bar{u}_{T_1} = n + 1, \quad \bar{v}_{T_1} = 0;$$

$$\bar{u}_{T_j} = n + 1, \quad \bar{v}_{T_j} = -(n + 1), \quad \text{for all } j = 2, \ldots, n - 1;$$

$$\bar{u}_{T_n} = -n, \quad \bar{v}_{T_n} = -(n + 1);$$

$$\bar{u}_{D_2} = \bar{v}_{D_2} = n;$$

$$\bar{u}_{T_j} = \bar{v}_{T_{2n+1-j}} \quad \text{for all } j = n + 1, \ldots, 2n; \quad \text{and}$$

$$\bar{v}_{T_j} = \bar{u}_{T_{2n+1-j}} \quad \text{for all } j = n + 1, \ldots, 2n.$$

Since $\bar{c}_{ij} = c_{ij} + \bar{u}_i + \bar{v}_j = 0$ for all $(i, j) \in A$ such that $\bar{x}_{ij} = 1$, $\bar{c}_{ij} = c_{ij} + \bar{u}_i + \bar{v}_j \geq 0$ otherwise, $(\bar{u}, \bar{v})$ is an optimal dual solution. The following reduced costs are null in the optimal solution of the assignment relaxation:

$$\bar{c}_{T_nD_1} = c_{T_nD_1} + \bar{u}_{T_n} + \bar{v}_{D_1} = 0;$$

$$\bar{c}_{D_2T_n} = c_{D_2T_n} + \bar{u}_{D_2} + \bar{v}_{T_n} = 0$$

$$\bar{c}_{T_{n+1}D_2} = c_{T_{n+1}D_2} + \bar{u}_{T_{n+1}} + \bar{v}_{D_2} = 0; \quad \text{and}$$

$$\bar{c}_{D_1T_{n+1}} = c_{D_1T_{n+1}} + \bar{u}_{D_1} + \bar{v}_{T_{n+1}} = 0.$$

Therefore, for every trip T_j, $j = 1, \ldots, 2n$, there is a path with null reduced costs leaving from each depot, visiting trip T_j (and possibly others), and coming back to the same depot, as shown in Figure 2. Hence, the

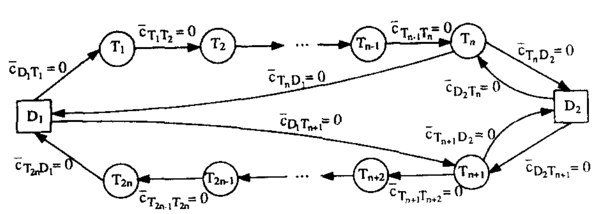

Figure 2. Paths with zero length in terms of the reduced costs.

shortest path relaxation does not improve the assignment bound and the additive bound is equal to the latter. Since the ratio $z_{\text{ADD}}/v(\mathbf{MDVSP})$ is null for this family of problems, this completes the proof that the additive bound can be arbitrarily bad.

Corollary 1. *The worst-case ratio $v(\mathbf{AP})/v(\mathbf{MDVSP})$ can be arbitrarily bad.*

Proof. From Theorem 3, since $v(\mathbf{AP}) \leq z_{\text{ADD}}$.

We now make the realistic assumption that whenever there is a path with travel time L from a depot D_k to the starting point of a trip T_j, there is also a shorter return path with a travel time less than or equal to L from the ending point of trip T_j to depot D_k. A graph with shorter return paths can be obtained, for instance, if the travel times in the street network are symmetrical (in particular, if every street segment can be traversed in both directions and with the same travel time) and the duration of each trip is included in the duration of its incoming arcs. With such arc durations, the cost of the optimal solution includes both productive travel times and nonproductive times between trips and between trips and depots. In most cities, even in the presence of one-way streets, the shorter return path assumption is satisfied in the large majority of the trips and, when it is violated, the return paths are not often too much larger than half of the overall path.

Theorem 5. *Under the shorter return path assumption, $v(\mathbf{AP})/v(\mathbf{MDVSP}) \geq 1/2$ and this bound is tight.*

Proof. A feasible solution to **MDVSP** can be built from the solution of the assignment relaxation by taking the latter and transforming each chain leaving from a depot D_{k_1}, performing trips $T_{j_1}, \ldots, T_{j_p}$, and arriving at depot D_{k_2}, into a cycle leaving from D_{k_1}, performing the same trips and coming back to D_{k_1} after T_{j_p}. Under the shorter return path assumption, the length of this cycle is at most twice the length of the original chain. Then, $v(\mathbf{MDVSP}) \leq 2 \cdot v(\mathbf{AP})$.

We now show that this bound is tight, i.e., it can be attained. We consider here a family $\{\Gamma^2(n)\}$ of instances of **MDVSP** with the same structure as $\{\Gamma^1(n)\}$, except for the travel times which are taken as: j, from depot D_k to the starting point of trip $T_{j+(k-1)n}$, $j = 1, \ldots, n$; j, from the ending point of trip $T_{j+(k-1)n}$, $j = 1, \ldots, n$ to depot D_k; $n + 1 - j$, from depot D_{k+1}(D_1, if $k = 2$) to the starting point of trip $T_{j+(k-1)n}$, $j = 1, \ldots, n$; $n + 1 - j$, from the ending point of trip $j = 1, \ldots, n$ to depot D_{k+1}(D_1, if $k = 2$); and $j - i$, from the ending point of trip $T_{i+(k-1)n}$ to the starting point of trip $T_{j+(k-1)n}$ for all $1 \leq i < j \leq n$. The optimal solutions to **MDVSP** and its assignment relaxation are the same as for the family $\{\Gamma^2(n)\}$. Then, $v(\mathbf{MDVSP}) = 2 \cdot (2n)$ and $v(\mathbf{AP}) = 2 \cdot (n+1)$. We now take the limit of the ratio $v(\mathbf{AP})/v(\mathbf{MDVSP})$ for the family $\{\Gamma^2(n)\}$ of problems:

$$\lim_{n\to\infty} v(\mathbf{AP})/v(\mathbf{MDVSP}) = \lim_{n\to\infty} 2 \cdot \frac{n+1}{2} \cdot (2n) = 1/2.$$

This completes the proof that under the shorter return path assumption the worst-case ratio 1/2 associated with the assignment bound is tight.

We will show that the same result holds for the additive bound. As for Theorem 4, the proof is based on the exhibition of a particular optimal dual solution to the assignment relaxation of every instance of $\{\Gamma^2(n)\}$, such that all shortest paths computed in terms of the reduced (residual) costs defined by this dual solution are null. Therefore, they do not improve the assignment bound.

Theorem 6. *Under the shorter return path assumption,* $z_{\text{ADD}}/v(\mathbf{MDVSP}) \geq 1/2$ *and this bound is tight.*

Proof. Take the following values for the dual variables u_i for all $i \in V$, and v_j for all $j \in V$:

$\bar{u}_{D_1} = \bar{v}_{D_1} = 0;$

$\bar{u}_{T_j} = -\bar{v}_{T_j} = j,$ for all $j = 1, 2, \ldots, n-1;$

$\bar{u}_{T_n} = \bar{v}_{T_n} = -n;$

$\bar{u}_{D_2} = \bar{v}_{D_2} = n-1;$

$\bar{u}_{T_j} = \bar{v}_{T_{2n+1-j}}$ for all $j = n+1, \ldots, 2n$; and

$\bar{v}_{T_j} = \bar{u}_{T_{2n+1-j}}$ for all $j = n+1, \ldots, 2n.$

Again, since $\bar{c}_{ij} = c_{ij} + \bar{u}_i + \bar{v}_j = 0$ for all $(i, j) \in A$ such that $\bar{x}_{ij} = 1$, $\bar{c}_{ij} = c_{ij} + \bar{u}_i + \bar{v}_j \geq 0$ otherwise, $(\bar{u}, \bar{v})$ is an optimal dual solution. The reduced costs $\bar{c}_{T_nD_1}$, $\bar{c}_{D_2T_n}$, $\bar{c}_{T_{n+1}D_2}$, and $\bar{c}_{D_1T_{n+1}}$ are null in the optimal solution of the assignment relaxation.

Therefore, as for Theorem 4, for every trip T_j, $j = 1, \ldots, 2n$, there is a path with null reduced costs leaving from each depot, visiting trip T_j (and possibly others), and coming back to the same depot. Hence, the additive bound is equal to the assignment bound. Since $v(\mathbf{MDVSP}) \leq 2 \cdot v(\mathbf{AP})$ under the shorter return path assumption and $v(\mathbf{AP}) \leq z_{\text{ADD}}$, this completes the proof that $z_{\text{ADD}}/v(\mathbf{MDVSP} \geq 1/2$ and this bound is tight.

6. CONCLUSIONS

We have given a new formulation to the multiple-depot vehicle scheduling problem as a set partitioning problem with side constraints, whose continuous relaxation can be solved by column generation. We have also established the relationship between the bounds obtained by the assignment relaxation, the shortest path relaxation, the additive technique, Lagrangian decomposition, and column generation. We have shown that the additive bound technique cannot provide tighter bounds than those obtained by Lagrangian decomposition, in the framework in which it has been used so far, and not better than the linear programming bound in the case of the **MDVSP**.

The column generation approach allows for efficiently solving the linear relaxation, without introducing other additional relaxations (e.g., the shortest path and assignment relaxations), as in the case of the additive bound technique. In fact, the column generation decomposition technique separates the flow constraints from the disjunctive constraints associated with the set partitioning formulation. This allows the use of efficient algorithms for solving the flow subproblems, leaving a small problem to be solved by linear programming. Moreover, the column generation bound simultaneously captures both the assignment and path constraints, contrarily to the additive bound technique which takes them into account one-at-a-time.

We have also shown that the column generation bound $v(\overline{\mathbf{MDVSP}})$ is at least as good as the additive bound z_{ADD}. The quality of the lower bounds provided by column generation is very good in practice and does not deteriorate with the increase in the number of trips or depots. We have exactly solved problems with up to 300 trips and 6 depots, which are four to five times larger than those solved on a previous work on a faster machine. The quality of the column generation bound is partly due to the structure of the set partitioning constraints (11) of **MDVSP′**. Let p_1, p_2, and p_3 be three columns, each associated with a circuit visiting two out of three trips T_a, T_b, and T_c in this order. Without loss of generality, suppose that p_1 visits T_a, then T_b, p_2 visits T_a, then T_c, and p_3 visits T_b then T_c. Since these three trips are pairwise compatible, there is also a circuit p_4 visiting T_a, next T_b, and then T_c. This special structure of the set partitioning constraints of **MDVSP′** eliminates many fractional solutions from its set of optimal solutions, and explains why the integrality gaps reported in the Section 4 are much smaller than those usually observed for other, more general set partitioning problems.

We also considered the worst-case behavior of lower bounds, with respect to the cost of the optimal integer solution. This is an interesting type of analysis, which has not received too much attention. In the case of the **MDVSP**, we studied the assignment bound, the shortest path bound, and the additive bound based on the assignment and shortest path relaxations. We have shown that this additive bound can be arbitrarily bad in the general case, and as bad as 50% in the symmetric case.

Concerning possible generalizations and extensions of this work, we notice that the software GENCOL is able to take into account additional constraints such as time windows, maximum capacity, precedence and coupling, duration, and resource consumption. Duration constraints are associated, e.g., to regulations concerning bus drivers, while length constraints are associated, e.g., to carburant consumption. Such additional constraints are dealt with by specialized shortest path algorithms for constrained shortest path problems during the column generation procedure. The approach proposed for the solution of the **MDVSP** could also be extended to scheduling problems with several types of vehicles. In that case, each depot would correspond to a different vehicle with a type-dependent cost.

ACKNOWLEDGMENT

The work of the first author was done during his sabbatical leave at GERAD and École Polytechnique de Montréal, and was partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico under grant 202005-89.5.

REFERENCES

Bertossi, A. A., P. Carraresi and G. Gallo. 1987. On Some Matching Problems Arising in Vehicle Scheduling Models. *Networks* **17,** 271–281.

Bodin, L., B. Golden, A. Assad and M. Ball. 1983. Routing and Scheduling of Vehicles and Crews: The State of the Art. *Comput. and Opns. Res.* **10,** 163–211.

Bodin, L., D. Rosenfield and A. Kydes. 1978. UCOST: A Micro Approach to a Transit Planning Problem. *J. Urban Anal.* **5,** 47–69.

Carpaneto, G., M. Dell'Amico, M. Fischetti and P. Toth. 1989. A Branch and Bound Algorithm for the Multiple Vehicle Scheduling Problem. *Networks* **19,** 531–548.

Carpaneto, G., M. Fischetti and P. Toth. 1989. New Lower Bounds for the Symmetric Travelling Salesman Problem. *Math. Prog.* **45,** 233–254.

Ceder, A., and H. I. Stern. 1981. Deficit Function Bus Scheduling With Deadheading Trip Insertions for Fleet Size Reduction. *Trans. Sci.* **15,** 338–363.

Dell'Amico, M., M. Fischetti and P. Toth. 1990. Heuristic Algorithms for the Multiple Depot Vehicle Scheduling Problem. DEIS, Technical Report OR/90/3, University of Bologna, Bologna, Italy.

Desrosiers, J., F. Soumis, and M. Desrochers. 1988. Lagrangian Relaxation Methods for Solving the Minimum Fleet Size Multiple Traveling Salesman Problem With Time Windows. *Mgmt. Sci.* **34,** 1005–1022.

El-Azm, A., 1985. The Minimum Fleet Size Problem and Its Applications to Bus Scheduling. In *Computer Scheduling of Public Transport 2,* J.-M. Rosseau (ed.). North-Holland, Amsterdam, 493–512.

Fischetti, M., and P. Toth. 1988. A New Dominance Procedure for Combinatorial Optimization Problems. *Opns. Res. Letts.* **7,** 181–187.

Fischetti, M., and P. Toth. 1989. An Additive Bounding Procedure for Combinatorial Optimization Problems. *Opns. Res.* **37,** 319–328.

Lasdon, L. S., 1970. *Optimization Theory for Large Systems.* MacMillan, New York.

Sansó, B., M. Desrochers, J. Desrosiers, Y. Dumas and F. Soumis. 1990. Modeling and Solving Routing and Scheduling Problems: GENCOL's Users Guide. Research Report, GERAD, Montreal, Canada.

Smith, B., and A. Wren. 1981. VAMPIRES and TASC: Two Successfully Applied Bus Scheduling Programs. In *Computer Scheduling of Public Transport: Urban Passenger Vehicle and Crew Scheduling,* A. Wren, (ed). North-Holland, Amsterdam, 97–124.