

M. MINOUX

**Optimal traffic assignment in a SS/TDMA frame : a new approach by set covering and column generation**

*RAIRO. Recherche opérationnelle*, tome 20, n° 4 (1986), p. 273-285

[http://www.numdam.org/item?id=RO\\_1986\\_\\_20\\_4\\_273\\_0](http://www.numdam.org/item?id=RO_1986__20_4_273_0)

© AFCET, 1986, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme  
Numérisation de documents anciens mathématiques  
<http://www.numdam.org/>

**OPTIMAL TRAFFIC ASSIGNMENT  
IN A SS/TDMA FRAME:  
A NEW APPROACH BY SET COVERING  
AND COLUMN GENERATION (\*)**

by M. MINOUX <sup>(2)</sup>

---

**Abstract.** — *We address here the problem of optimal assignment of a given traffic matrix, within a satellite system TDMA (time division multiple access) frame where no traffic splitting is allowed (in view of keeping the number of different switching states as low as possible). After reformulating the problem as a large scale set covering problem, it is shown that, in spite of the large number of columns, its continuous relaxation can be solved optimally according to a column-generation procedure (generalized linear programming). Computational experiments show that this approach usually produces lower bounds which significantly improve upon the trivial lower bounds used so far for this problem (the greatest row or column sum of the traffic matrix) thus opening the way to exact solution of problems of higher dimensions.*

**Keywords:** Combinatorial optimization; generalized linear programming; column generation; assignment problem.

**Résumé.** — *On s'intéresse ici au problème d'affecter une matrice de trafic donnée dans une trame satellite AMRT (accès multiple à répartition dans le temps) lorsque le fractionnement du trafic n'est pas autorisé. Après avoir reformulé le problème comme un problème de recouvrement de grande dimension, on montre qu'en dépit du très grand nombre de colonnes, la relaxation continue peut être résolue de façon exacte par programmation linéaire généralisée (génération de colonnes). Les résultats de calcul obtenus montrent que cette approche produit des bornes inférieures qui améliorent significativement les bornes élémentaires utilisées jusqu'à présent (la plus grande somme en ligne ou en colonne de la matrice de trafic) et qui sont même très souvent optimales. Elle ouvre donc la voie à la résolution exacte, en nombres entiers, de problèmes de dimensions plus élevées.*

**Mots clés :** optimisation combinatoire; programmation linéaire généralisée; génération de colonnes; affectation.

---

(\*) Received March 1986.

Paper presented at the ORSA/TIMS meeting, Dallas, Texas, Nov. 26-28 1984.

<sup>(2)</sup> LAMSADE, Université Paris-IX - Dauphine, 1, place Maréchal-de-Lattre-de-Tassigny, 75775 Paris Cedex.

## 1. INTRODUCTION

On-board switching systems for satellite communications, which are based on the *Time Division Multiple Access principle* (TDMA), are used to set up, periodically and at specific instants (to be determined) digital radio channels between a number of *earth stations*. The total switching capacity of the satellite system is expressed in terms of the number of elementary time periods contained in the TDMA *frame* (in practice this number is about 1.000 to 2.000). An elementary time period in the frame corresponding to *one digital voice channel* each interstation traffic value is expressed as a number of elementary time periods (i. e. as a number of required digital voice channels).

A fundamental problem arising in this context, is to find an optimum TDMA frame time schedule, in other words, to find an optimal "timetable" for the switching system on board. Though this kind of problem may be formulated in many ways, it is often stated as follows: *Given an interstation traffic matrix, determine the successive switching modes (and in particular, the very instants when switching states should be changed) in order to switch all the traffic requirements in minimum time.*

A simplified version of this problem, where it is assumed that the traffic blocks can be split (divided into subblocks) as much as desired, has first been studied by Ito *et al.* (1977) who have shown that in that case there always exists an optimal schedule with duration equal to the largest row or column sum in the traffic matrix. The method suggested to actually build such an optimal schedule was based on a *greedy-type algorithm* (however incorrect as pointed out by Inukai 1978). Inukai (1979) has later on proposed a correct greedy solution method for the same problem. However, one of the results of this work showed that, when an optimal schedule (with duration equal to the largest row or column sum) is required, then the number of different switching modes may, at least in the worst case, be as large as  $n^2 - 2n + 2$  for a  $n \times n$  traffic matrix. On real problems, the observed figures are usually not that large, but may stay rather high (cf. the computational experiments in Bongiovanni *et al.* 1981). In practice, any change in the switching mode requiring transmission of some signalling information (preambles) to be inserted within the frame, such results are usually not directly applicable.

This explains why more realistic (though more complex) models have been subsequently studied, in which a good compromise between the length of the schedule in the frame and the number of necessary different switching modes is looked for. Along this line, Gopal and Wong (1983) have proposed a method aiming at explicitly minimizing the number of different switching modes. Natarajan and Calo (1980) suggest a number of heuristic methods

for minimizing the length of the schedule under the constraint that the number of distinct switching modes be *minimal* (equal to  $n$ , the size of the traffic matrix in case of a full density matrix). However such a constraint may prove in practice to be uselessly strong (relaxing this constraint only slightly, e. g. by allowing just  $n+1$  or  $n+2$  switching modes may lead to significantly shorter schedules). That is why it appears that the most extensively studied model consists in minimizing the length of the schedule in the frame, while *indirectly* limiting the number of switching modes by forbidding any splitting of the interstation traffic blocks.

For this difficult combinatorial optimization problem (as a scheduling problem with pure disjunctive constraints and no precedence constraint, it is *NP-Hard* as shown by Gonzales and Sahni 1976) quite a lot of heuristics have been proposed: Camerini, Maffioli and Tartara (1981), Gopal (1982), Balas and Landweer (1983). On the contrary, very few attempts seem to have been made towards the *exact solution* of this problem (e. g. by Branch and Bound techniques). The reason of this is most likely to be found in the *poor quality* of the lower bounding functions which have been known so far. As an example of such an attempt, the work by L. Vismara (together with P. Camerini and F. Maffioli) which uses as lower bounds, the largest row or column sum of the matrix, only allowed to get exact optimal solutions on very small sized examples ( $5 \times 5$  or  $6 \times 6$ ).

We propose here a new approach based on reformulating the problem as a *set covering problem with a huge number of variables*. It is shown how the continuous relaxation of this problem may nevertheless be solved to optimality by using a *column generation technique* (or generalized linear programming), the determination of a minimum reduced cost column being achieved through the solution of a sequence of ordinary *assignment problems*.

Preliminary computational testing shows that this solution procedure systematically produces lower bounds which not only significantly improve upon the previously known bounds (largest row or column sum) but are almost always *optimal*, thus confirming the relevance of the new approach. Current work is going on towards exploiting the various informations provided by the optimal solutions to the continuous set covering problem within tree search methods (Branch and Bound).

## 2. OPTIMAL DECOMPOSITION OF A MATRIX INTO SWITCHING MODE MATRICES

Let  $T = (t_{ij})_{\substack{i=1 \dots n \\ j=1 \dots n}}$  be a square  $n \times n$  matrix (traffic matrix) with non negative entries.

We address here the problem of finding a *decomposition* of  $T$  into:

$$T = \sum_{k=1}^K P^k \quad (1)$$

with the following conditions:

(i)  $\forall k = 1, \dots, K, P^k = (P^k_{ij})_{\substack{i=1 \dots n \\ j=1 \dots n}}$  is a *switching mode matrix*, i. e.

a matrix containing no more than one nonzero entry in each row or column.

(ii)  $\forall k = 1, \dots, K, P^k_{ij} > 0 \Rightarrow P^k_{ij} = t_{ij}$  (thus each entry of the original matrix will appear in one and only one matrix of the decomposition).

(iii) The sum  $\sum_{k=1}^K |P^k|$  is *minimized*, where for any  $n \times n$  matrix  $Q = (q_{ij})$ .

$|Q|$  denotes the  $L_\infty$  norm of matrix  $Q$  i. e.:

$$|Q| = \text{Max}_{(i,j)} \{ |q_{ij}| \}.$$

Observe that, in the above problem, the number  $K$  of mode matrices in the decomposition is *not imposed*: this number should actually be the result of the optimization process. However, it is always possible to refer to the case of fixed  $K$  in view of the following remark.

If  $M (M \leq n^2)$  is the number of nonzero entries in  $T$ , any decomposition of type (1) will contain at most  $M$  mode matrices. Thus one may choose  $K=M$  in (1) by considering that some of the matrices  $P^k$  will be zero (when  $P^k=0$ , then  $|P^k|=0$ , hence such matrices do not play any role in the objective function).

### 3. FORMULATION AS A LARGE SCALE SET COVERING PROBLEM

We show in this section that the search for an optimum decomposition of a given matrix into switching mode matrices can be formulated as a set covering problem in 0-1 variables but with a very large number of columns.

If  $M$  denotes, as in section 2, the number of nonzero entries of the given matrix, each switching mode matrix  $P^j$  which may be involved in the decomposition (1) can be characterized by a 0-1 vector  $A^j = (a_{ij}) \ i = 1 \dots M$  such that:

$a_{ij} = 1$  if  $P^j$  contains the  $i$ th nonzero entry of matrix  $T$   
 $= 0$  otherwise.

There are as many distinct mode matrices as there are ways of choosing 1, 2, ...,  $n$  independent nonzero entries in  $T$  (a subset of entries is called *independent* if and only if it contains at most one entry in each row or column). We shall denote by  $N$  the total number of distinct switching mode matrices (obviously  $N < 2^M$ ), and  $A = (a_{ij})$  the  $M \times N$  0-1 matrix whose columns are the characteristic vectors of the independent subsets of nonzero entries of  $T$ .

With each column  $j$  of  $A$  (with each mode matrix) we associate a binary variable  $x_j$  such that:

$$x_j = 1 \iff \text{mode matrix } j \text{ is selected in the solution}$$

and a cost  $c_j$  equal to the value of the largest entry of mode matrix  $j$ .

The problem can then be formulated as the search for a minimum cost covering of the  $M$  entries of  $T$ , i. e. as the following *set covering problem*:

$$(SC) \left\{ \begin{array}{l} \text{Minimize } \sum_{j=1}^N c_j x_j \\ \text{subject to} \\ A \cdot x \geq \mathbf{1} \\ x \in \{0, 1\}^N \end{array} \right.$$

where  $\mathbf{1}$  denotes the  $M$ -vector with all components equal to 1.

*An example*

Consider the  $3 \times 3$  matrix below, which contains 6 nonzero entries numbered from 1 to 6 (in parentheses):

$$T = \begin{bmatrix} 2_{(1)} & 0 & 7_{(2)} \\ 5_{(3)} & 4_{(4)} & 0 \\ 0 & 1_{(5)} & 9_{(6)} \end{bmatrix}$$

The corresponding set covering problem for this example is stated below and includes 18 columns (18 distinct independent subsets of nonzero entries):

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$	$x_{11}$	$x_{12}$	$x_{13}$	$x_{14}$	$x_{15}$	$x_{16}$	$x_{17}$	$x_{18}$
costs →	9	7	7	2	4	9	7	7	7	5	9	9	2	7	5	4	1	9
1	1			1	1	1							1					
2		1	1				1	1	1					1				
3		1					1			1	1				1			
4	1		1		1			1				1				1		
5		1		1					1	1							1	
6	1					1					1	1						1

independent  
subsets with  
3 elements

independent subsets  
with 2 elements

independent  
subsets with  
1 element

The optimal integer solution to this problem is  $x_1^* = 1, x_2^* = 1$  (all the other variables being 0) with optimal cost equal to 16.

This optimal solution corresponds to the optimal decomposition of  $T$ :

$$T = \begin{pmatrix} 2 & 0 & 7 \\ 5 & 4 & 0 \\ 0 & 1 & 9 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 9 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 7 \\ 5 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

**4. SOLUTION OF THE CONTINUOUS RELAXATION OF (SC) BY COLUMN GENERATION. LOWER BOUNDS**

Balas and Landweer (1983) used the set covering formulation (SC) to devise efficient heuristic procedures to obtain good approximate solutions to the optimal matrix decomposition problem. We propose here to exploit this formulation in another way, namely the exact solution of the continuous relaxation of this problem by generalized linear programming techniques.

Due to the huge number of variables there is no hope of solving directly (SC) as an integer programming problem.

However, solving the continuous relaxation of (SC) may be of interest, at least because it will provide *lower bounds* to the optimum integer solution

cost. We shall denote by  $(\overline{SC})$  the (large scale) linear program obtained by relaxing the integrality requirements in (SC), i. e.:

$$(\overline{SC}) \left\{ \begin{array}{l} \text{Minimize } \sum_{j=1}^N c_j x_j \\ \text{s. t.} \\ Ax \geq \mathbf{1} \\ x \in [0, 1]^N. \end{array} \right.$$

The above linear program has only a limited number of constraints ( $M \leq n^2$ ) but a huge number of columns as soon as the size of the matrix exceeds 7 or 8. In practical applications, one may have  $n \simeq 10$  to 50, which definitely excludes the possibility of explicitly stating the full matrix  $A$ .

In that situation however, there exists a particular technique, known as *generalized linear programming* or *column generation* (cf. Dantzig 1963 for instance) which allows solving such large scale linear programs, under the mere condition that, at each step, there is an appropriate method (more efficient than enumeration) to find out a *minimum reduced cost column* (the so-called “pricing out” process). Recall that the *reduced cost*  $\bar{c}_j$  of any column  $A^j$  is defined by:

$$\bar{c}_j = c_j - \pi \cdot A^j$$

where  $\pi = (\pi_1, \dots, \pi_M)$  is the current vector of *simplex multipliers*.

For the relaxed set covering problem  $(\overline{SC})$ , we now show that this *column generation problem* can be efficiently solved by a *polynomial time algorithm*.

Denote by  $\mathcal{J} \subset \{0, 1\}^M$  the set of all independent subsets of nonzero entries in  $T$ .  $\pi = (\pi_1, \dots, \pi_M)$  being, as above, the current simplex multipliers, the problem is to determine the independent subset  $\bar{v} = (\bar{v}_l)_{l=1, \dots, M} \in \mathcal{J}$  such that:

$$c(\bar{v}) - \sum_{l=1}^M \bar{v}_l \pi_l = \text{Min}_{v \in \mathcal{J}} \left\{ c(v) - \sum_{l=1}^M v_l \cdot \pi_l \right\} \tag{2}$$

where,  $\forall v \in \mathcal{J}$ ,  $c(v)$  is the cost of the independent subset  $v$ , i. e. the value of the largest entry in the corresponding mode matrix.

Though much alike an assignment problem, this problem cannot be solved as it stands, as an ordinary (min-sum) assignment problem. As a matter-of-fact, it is seen that the objective function in (2) is a combination of a min-max criterion [term  $c(v)$ ] and a min-sum criterion (term  $-\sum v_l \pi_l$ ). However, it can be observed that  $c(v)$  can take on at most  $M$  distinct values, corresponding



to the  $M$  nonzero entries of matrix  $T$ . Hence the idea of solving (2) via a *parametric approach* where each of the possible values of  $c(v)$  is examined in turn.

Suppose first that we know the value  $c(\bar{v})$  in (2): this means that we know the largest entry of matrix  $T$  involved in  $\bar{v}$ . Problem (2) then reduces to the following problem (3) for the value  $\alpha = c(\bar{v})$  of parameter  $\alpha$

$$\left. \begin{array}{l} \text{Determine an independent subset of weight } \sum_{i=1}^M \pi_i v_i \text{ maximum in} \\ \mathcal{I}(\alpha), \text{ where } \mathcal{I}(\alpha) \text{ denotes the set of all independent subsets of} \\ \text{entries of matrix } T \text{ which have value } \leq \alpha. \end{array} \right\} \quad (3)$$

Since the value  $c(\bar{v})$  is not known *a priori*, it will thus be necessary to solve (3) for each of the possible values of  $\alpha$ , the number of which is less than or equal to  $M$  (number of entries with *distinct* values in matrix  $T$ ).

Now for any value of  $\alpha$ , problem (3) is recognized as a classical assignment problem, which can be solved as a maximum cost network flow problem between  $s$  and  $t$  in the following transportation network  $G=[X, U]$ :

– The set of nodes  $X$  is of the form  $R \cup C \cup \{s, t\}$  where  $R$  corresponds to the  $n$  rows of  $T$ ,  $C$  to the  $n$  columns of  $T$ , and where  $s$  and  $t$  are two additional vertices called the *source* and the *sink* respectively.

– The set of arcs  $U$  is formed by:  $n$  arcs of the form  $(s, r)$  ( $r \in R$ ) with capacity 1 and zero cost;  $n$  arcs of the form  $(c, t)$  ( $c \in C$ ) with capacity 1 and zero cost; as many arcs of the form  $(r, c)$  (where  $r \in R$  and  $c \in C$ ) as there are entries  $t_{rc} \leq \alpha$  in  $T$ . With each such arc  $(r, c)$  we associate infinite capacity and cost equal to  $\pi_i$ , the simplex multiplier attached to the entry  $t_{rc}$  of matrix  $T$ .

If this problem is solved by using the Busacker and Gowen algorithm (1961) properly improved by the Edmonds and Karp technique (1972), the complexity is  $O(n^3)$  for each single value of  $\alpha$ , and since there are at most  $M$  distinct possible values of  $\alpha$  to consider, the overall complexity for solving the column generation subproblem would be  $O(Mn^3)$ .

However, substantial improvement can still be gained as the following result shows.

**THEOREM 1.** — *The column generation subproblem (2) can be solved in worst case polynomial time  $O(Mn^2)$  where  $M$  is the number of nonzero entries in the given traffic matrix ( $M \leq n^2$ ).*

*Proof.* — Start by solving problem (3) on  $\mathcal{S}(\alpha_{\min})$  where  $\alpha_{\min}$  is the value of the *least entry* in  $T$ ; this amounts to solving an assignment problem on a  $n \times n$  auxiliary matrix with only one nonzero entry corresponding to the least term in  $T$ . Then raising  $\alpha$  little by little results in incorporating in the auxiliary matrix new entries, one at a time, at places corresponding to the entries of  $T$  taken in *increasing order*. Each time a new entry is incorporated one checks whether the current assignment is still optimal or not. This can be efficiently tested in time complexity  $O(1)$  by computing the *reduced cost* of the entry under consideration (since the dual variables of the assignment problem are available in the Busacker and Gowen-Edmonds and Karp procedure),

If the reduced cost is non positive (remember that we are maximizing  $\sum \pi_i v_i$ ) then the current assignment remains optimal, and we just have to proceed to the next candidate entry. If the reduced cost is strictly positive, then this means that the current assignment is no longer optimal, and we have to perform a *flow change*. Now, since we are certain that the new optimal assignment *should* actually contain the new entry, this flow change consists in sending one additional unit of flow on the circuit of maximum reduced cost running through the arc associated with the new entry in the so-called *incremental graph* (see Gondran and Minoux 1979, 1984). Reoptimizing thus only requires *one longest path calculation* in a  $n \times n$  bipartite graph, the lengths on the arcs being nonpositive. Since Dijkstra's algorithm solves this problem in  $O(n^2)$  time complexity, and the maximum number of necessary flow changes is bounded by  $M$  in the worst case, the theorem follows.  $\square$

## 5. PRELIMINARY COMPUTATIONAL RESULTS AND CONCLUSIONS

The column generation algorithm has been implemented and preliminary computational tests were carried out on a number of small-to-medium sized problems (for  $n$  ranging from 5 to 8). The numerical results obtained are subsumed in table I below where the following indications are displayed:

- The reference number of the example together with its size. The label (R) means that the example corresponds to a real traffic matrix, the label (NR) means that it corresponds to a fictitious one (usually, randomly generated).

- $M$ , the number of nonzero entries in the traffic matrix.

- $z_0$ , the value of the starting solution; in all but one examples, the starting solution consists in  $M$  switching mode matrices ( $M$  columns), each containing exactly one of the nonzero entries of the original traffic matrix. In Example 5 only, the solution provided by the greedy heuristic (and

TABLE I

Results of computational experiments with the column generation algorithm.

Example	$M$ nb of terms	$z_0$ starting solution	Nb col. in final solution	$z_c^*$ contin. optimum	Structure of contin. solution	$z_G$ Greedy heuristic	$Z_L$ lower bound	$Z_I^*$ integer optimum
EX1 $5 \times 5$ (NR) . . . .	16	1.528 (16 cols)	16+46 (22 iter.)	565 (integer)	5 variables set to "1"	585	555	565
EX2 $6 \times 6$ (R) . . . . .	18	7.180 (18 cols)	18+55 (14 it.)	3.380	3 Var. "1" 4 var. "0.5"	3.380	3.380	3.380
EX3 $6 \times 6$ (NR) . . . .	26	2.490 (26 cols)	26+120 (26 it.)	650	3 Var. "1" 4 var "0.5"	692	605	650
EX4 $5 \times 5$ (NR) . . . .	25	11.857 (25 cols)	25+52	3.388	1 Var. "1" 8 var. "0.5"	3.496	2.867	3.388
EX5 same as EX4 . . . .	25	3.496 (6 cols) greedy sol.	6+60	3.388	2 Var. "1" 6 var. "0.5"	3.496	2.867	3.388
EX6 $6 \times 6$ (R) . . . . .	16	9.168 (16 cols)	16+40 (9 it.)	3.983 (integer)	4 Var. "1"	3.983	3.983	3.983
EX7 $8 \times 8$ (R) . . . . .	36	8.660 (36 cols)	36+128 (16 it.)	3.710	19 var. fractional (0.2; 0.4; 0.6)	3.710	3.710	3.710
EX8 $7 \times 7$ (NR) . . . .	34	1.482 (34 cols)	34+153 (20 it.)	341 (integer)	6 Var. "1"	383	325	341
EX9 $7 \times 7$ (NR) . . . .	26	5.738 (26 cols)	26+130 (27 it.)	1.154 (integer)	5 Var. "1"	1.154	1.055	1.154
EX10 $8 \times 8$ (NR) . . . .	36	7.372 (36 cols)	36+252 (38 it.)	1.360 (integer)	5 Var. "1"	1.627	1.254	1.360
EX11 $8 \times 8$ (NR) . . . .	34	7.560 (34 cols)	34+204 (38 it.)	1.298 (integer)	5 Var. "1"	1.420	1.228	1.298

containing only 6 switching mode matrices) was taken as starting solution (indeed, the only difference between Examples 4 and 5 lies in the starting solution).

– The total number of columns added to the initial set of columns by the column generation process, in order to reach the continuous optimal solution. When multiple pricing was used (several columns being selected each time the column generation algorithm was applied) we indicate in parentheses the number of times the column generator was called for.

–  $z_c^*$ , the value of the continuous optimal solution.

– Some indications about the structure of the continuous optimal solution (number of variables with value 1, number of fractional variables).

–  $z_G$ , the value of an approximate solution obtained through the *greedy heuristic* which consists in building successive switching mode matrices in the following way. Suppose that mode matrices 1, 2, ...,  $k-1$  have already been built and consider the residual traffic matrix consisting in those entries which have not been assigned to a switching mode matrix yet. The  $k$ th mode matrix is then determined by selecting as many entries of the residual matrix as possible, according to the rule that, at each step, the largest entry which does not conflict with the previously selected entries is chosen. (In spite of its simplicity, this heuristic procedure exhibits pretty good average performance in terms of the quality of the solution costs obtained, as already observed by several authors; see e. g. the comparative computational results presented in Brandt 1982.)

–  $z_L$  the lower bound to the optimal integer solution cost equal to the largest row or column sum of the traffic matrix.

–  $z_I^*$  the cost of an optimal integer solution (obtained by a branch and bound procedure).

Various interesting observations can be drawn from Table I.

(a) 6 out of 10 problems (recall that Examples 4 and 5 concern the same problem) lead to an optimal integral solution, and this not only occurred in “easier cases” (when the greedy heuristic happened to provide the exact optimal solutions too, see EX6, EX9) but also in some “harder cases” (when the greedy solution proved to be far away from the exact optimal solution, and the lower bound  $z_L$  was observed to be rather poor, see EX1, EX8, EX10, EX11). This is a highly positive result since, if we consider the problems of larger sizes among the “harder cases” (i. e. EX8, EX10 and EX11), it can be easily realized that no existing exact (branch and bound) method based on the lower bound  $z_L$  could have solved them to optimality (see the already

mentioned computational results of Vismara 1982), whereas the column generation method provided guaranteed optimal integer solutions.

(b) The problems corresponding to real data (labelled "R" in first column of table I), at least those dealt with here, appeared to be in the "easier cases" where both the greedy solution value and the lower bound were equal (EX2, EX6, EX7). Somewhat surprisingly, in those cases, the column generation method (in addition to being obviously more time-consuming) provided *fractional solutions* in 2 over 3 of the examples (EX2, EX7). An explanation of this is that, in such situations, very high degeneracy occurs (the continuous optimum solution set is large) thus the probability of getting an integer point is low. In practice, this strongly suggests to restrict the use of the column generation technique to the "harder cases" where, after applying the greedy algorithm (or any possibly more involved heuristic) a *gap* is observed between  $z_G$  (the heuristic solution obtained) and  $z_L$  (the lower bound).

(c) In all the examples treated, the lower bound  $z_c^*$  appears to be *optimal*, i. e. exactly equal to the cost of an integral solution. This clearly demonstrates the interest of the approach presented here since, even if the existence of a *gap* between  $z_c^*$  and  $z_I^*$  for some special instances of the problem cannot be ruled out, at least significant improvements upon the usual lower bound can be expected (note that the relative improvement  $(z_c^* - z_L)/z_L$  is about 7,5% for EX3 and as large as 17% for EX4-5). This suggests that, incorporating the column generation scheme for getting lowers bounds (and possibly helping in providing better feasible integer solutions than those provided by known existing heuristics), in the framework of a branch and bound process, should be worth while future investigation, and is likely to open the way to exact solution, even in the harder cases, of problems of significantly higher dimensions.

#### ACKNOWLEDGMENTS

Help of M. C. Barnaud in the computational testing of the column generation method is gratefully acknowledged.

#### REFERENCES

- E. BALAS and P. R. LANDWEER, *Traffic Assignment in Communication Satellites*, Operations Research Letters, Vol. 2, No. 4, 1983, pp. 141-147.
- G. BONGIOVANNI, D. COPPERSMITH and C. K. WONG, *An Optimum Time Slot Assignment Algorithm for an SS/TDMA System with Variable Number of Transponders*, IEEE Transactions on Communications, vol. 29, No. 5, 1981, pp. 721-726.
- G. BONGIOVANNI, D. T. TANG and C. K. WONG, *A General Multibeam Satellite Switching Algorithm*, IEEE Transactions on Communications, vol. 29, No. 7, 1981, pp. 1025-1036.

- E. BRANDT, *Optimisation de ressources pour des transmissions par satellites utilisant l'AMRT*. Thèse 3<sup>e</sup> cycle, Université de Paris-Sud, Orsay, 1982.
- R. G. BUSACKER and P. J. GOWEN, *A Procedure for Determining a Family of Minimal Cost Network Flow Patterns*, O.R.O. Technical Report 15, Johns Hopkins University, 1961.
- P. CAMERINI, F. MAFFIOLI and G. TARTARA, *Some Scheduling Algorithms for SS/TDMA Systems*, Proc. Vth Internat Conf. on Digital Satellite Comm., 1981, Genoa, Italy.
- G. B. DANTZIG, *Linear Programming and Extensions*, Princeton University Press, 1963.
- J. EDMONDS and R. M. KARP, *Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems*, Journal ACM, Vol. 19, No. 2, 1972, pp. 248-264.
- L. R. FORD and D. R. FULKERSON, *Flows in Networks*, Princeton University Press, 1962.
- M. GONDRAN and M. MINOUX, *Graphes et Algorithmes*, Eyrolles, Paris, 1979. English translation J. Wiley & Sons 1984.
- T. GONZALES and S. SAHNI, *Open Shop Scheduling to Minimize Finish Time*, J. ACM, Vol. 23, 1976, pp. 665-679.
- I. GOPAL and C. K. WONG, *Minimizing the number of switchings in an SS/TDMA system*, Proceedings Intern. Symp., *Satellite and Computer Communications*, pp. 27-29, 1983, Versailles, France.
- I. S. GOPAL, *Scheduling Algorithms for Multi-Beam Communication Satellites*, Ph. D. Dissertation, Columbia University, New York, NY, 1982.
- I. S. GOPAL, G. BONGIOVANNI, M. A. BONUCELLI, D. T. TANG and C. K. WONG, *An Optimal Switching Algorithm for Multibeam Satellite Systems with Variable Bandwidth Beams*, IEEE Transactions on Communications, Vol. 30, No. 11, 1982, pp. 2475-2481.
- I. S. GOPAL, M. A. BONUCELLI and C. K. WONG, *Scheduling in Multibeam Satellites with Interfering Zones*, IEEE Transactions on Communications, Vol. 31, No. 8, 1983, pp. 941-951.
- I. S. GOPAL, D. COPPERSMITH and C. K. WONG, *Minimizing Packet Waiting Time in a Multibeam Satellite System*, IEEE Transactions on Communications, Vol. 30, No. 2, 1982, pp. 305-316.
- T. INUKAI, *Comments on Analysis of a Switch Matrix for an SS/TDMA System*, Proceedings of the IEEE, Vol. 66, No. 12, 1978, pp. 1669-1670.
- T. INUKAI, *An Efficient SS/TDMA Time Slot Assignment Algorithm*, IEEE Transactions on Communications, Vol. 27, No. 10, 1979, pp. 1449-1455.
- Y. ITO, Y. URANO, T. MURATANI and M. YAMAGUCHI, *Analysis of a Switch Matrix for an SS/TDMA System*, Proceedings of the IEEE, Vol. 65, No. 3, 1977, pp. 411-419.
- M. MINOUX, *Programmation Mathématique* (tomes 1 et 2). Collection Technique et Scientifique des Télécommunications, Dunod, Paris, 1983. English translations J. Wiley & Sons 1986.
- NATARAJAN and CALO, *Time Slot Assignment in a SS/TDMA System with minimum Switchings*, Report RC 8342, 7/2/80 IBM Thomas J. Watson Research Center. Yorktown Heights, NY 10598, 1980.
- L. VISMARA, *Piano di accesso dei messaggi in sistemi SS/TDMA: un metodo di enumerazione implicita per minimizzare il tempo di trasmissione*, Tesi di laurea, Politecnico di Milano, Depto di Elettronica. (Paolo Camerini i Francesco Maffioli Relatori), 1982.