



**HAL**  
open science

## Quasi-optimal bandwidth allocation for multi-spot MFTDMA satellites

Sara Alouf, Eitan Altman, Jérôme Galtier, Jean-François Lalande, Corinne  
Touati

► **To cite this version:**

Sara Alouf, Eitan Altman, Jérôme Galtier, Jean-François Lalande, Corinne Touati. Quasi-optimal bandwidth allocation for multi-spot MFTDMA satellites. INFOCOM 2005, Mar 2005, Miami, United States. pp.560-571, 10.1109/INFOCOM.2005.1497923 . inria-00451816

**HAL Id: inria-00451816**

**<https://inria.hal.science/inria-00451816>**

Submitted on 12 Jul 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Quasi-optimal bandwidth allocation for multi-spot MFTDMA satellites

Sara Alouf<sup>◇\*</sup>, Eitan Altman<sup>◇</sup>, Jérôme Galtier<sup>\*,◇,†</sup>, Jean-François Lalande<sup>◇</sup> and Corinne Touati<sup>◇</sup>

## Abstract

This paper presents an algorithm for resource allocation in satellite networks<sup>1</sup>. It deals with planning a time/frequency plan for a set of terminals with a known geometric configuration under interference constraints. Our objective is to maximize the system throughput while guaranteeing that the different types of demands are satisfied, each type using a different amount of bandwidth. The proposed algorithm relies on two main techniques. The first generates admissible configurations for the interference constraints, whereas the second uses linear and integer programming with column generation. The obtained solution estimates a possible allocation plan with optimality guarantees, and highlights the frequency interferences which degrade the construction of good solutions. **Keywords:** Combinatorics, mathematical programming/optimization, system design.

## 1 Introduction

We consider a multi-spot geostationary satellite system for which a manager assigns satellite uplink MFTDMA (Multi-Frequency Time-Division Multiple Access) slots to service providers (operators). The service providers themselves operate a park of terminals distributed on the satellite area of cover. Concerning the radio channel, the satellite divides the time and frequency spectrum into time slots. Geographically, the terminals are distributed on zones, themselves being included in *spots*, which correspond to equipments of reception (beams) of the satellite. Radio interferences impose constraints on the slots that can simultaneously be assigned in different spots that have the same frequency. A slot cannot be assigned simultaneously to more than one zone in a spot. Spots are given colors (bands of frequencies) and spots of different colors do not interfere, but spots of the same color do, and a slot can be assigned to an operator in a given zone only if the interference it experiences with the other active zones is below a given threshold. Slot assignment is static but can be changed once per hour (due to changes in demands, on the one hand, and to changes in atmospheric conditions, on the other hand). Every hour, the demand of the service providers is re-evaluated and a new allocation could be generated. Due to real-time constraints, solutions are needed within a few minutes.

Our goal is to maximize the throughput of the system. The approach adopted to achieve this goal can be formulated as a fractional coloring problem [11]. Casting the problem into coloring

of graphs shows that it is NP-complete to maximize the throughput [7, GT20]. Instead, we propose to solve the problem using a linear and integer programming approach with column generation.

This work is clearly motivated by the cost of the design of satellite antennas [2]. The cost of an antenna is a strong function of its size, roughly speaking, proportional to the diameter cubed. Larger antennas generate small interferences and have better gain, but increase tremendously the cost of the satellite. One of the goals of this approach is to tune precisely the assignment problem given its profile in terms of interference and gain. We will see that in return, our program can derive which interferences are responsible for (sometimes substantial) loss of capacity for a given demand.

In our experiments to evaluate the proposed approach, we will be using two series of data corresponding to 8 and 32 spots per color respectively. We assumed that there are three zones per spot, and four types of carriers<sup>2</sup>. Our work is focused on one of the colors of the bandwidth (recall that spots of different colors do not interfere with each other), so that the complete processing phase should use the same program for each color (if necessary in a parallel way). In our experiments, the total number of time slots that can be assigned is set to 3456.

We propose in this paper a linear and integer programming approach that allows to solve the problem almost optimally. For the 8-spot case, the problem is solved in a minute or so, with a guarantee of consuming at most 1% more bandwidth than the absolute optimum. The dual/primal approach is exploited in a master/slave fashion, where the master program is a heuristic that finds non-interfering zones that are directly translated into valid columns for the primal problem handled by the slave program. This approach can output the interfering configurations that limit the optimization up to a certain threshold. This information is extremely important for the design of antennas since it explains the characteristics of the antennas that lead to performance limitation. In other words, our approach identifies the interfering configurations that are crucial to the optimization, and this information has to be taken into account when designing antennas. Designers have to make sure that the antennas do not impair such configurations. Last, we show that, in the 32-spot case, our program can output solutions that in practice have good performance.

Due to lack of space we do not discuss in details related references which have appeared in the past; they all dealt with simpler models that in some cases have been solvable using polynomial algorithms. We refer to the book chapter [1] for a survey. We wish to mention however that problems with similar nature but with

<sup>◇</sup> INRIA Sophia Antipolis – B.P. 93 – 06902, Sophia Antipolis Cedex, France – {salouf, altman, jgaltier, jflaland, ctouati}@sophia.inria.fr

<sup>†</sup> France Telecom R&D

<sup>1</sup> This work is part of research convention A 56918 between INRIA and AL-CATEL SPACE INDUSTRIES (contract number 1 02 E 0306 00 41620 01 2).

<sup>2</sup> Carriers have different bandwidths thus providing different slot durations. The use of a specific carrier by a given terminal is determined by the terminal's transmission capability.

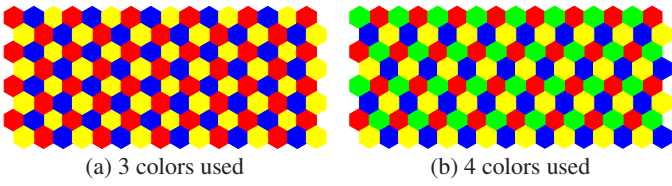


Figure 1: Spatial distribution of spots and optimal reuse of colors.

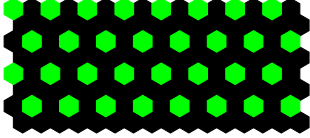


Figure 2: Spatial distribution of spots using the same color (4 colors case).

simpler structure have also been treated in the context of scheduling in ad-hoc networks, see e.g. [8] and references therein.

The structure of the paper is as follows. The system model and its constraints are presented in Section 2. The resolution of the time slot allocation problem throughout a simple example is detailed in Section 3, whereas the general solution is detailed in Section 4. Numerical results are presented in Section 5, followed by a concluding section.

## 2 The model

### 2.1 Spatial reuse

The total satellite bandwidth is subdivided in several equally-large bandwidths. Each one of these will be assigned a *color*. Every spot is assigned a unique fixed color, implying that all terminals of a spot can transmit within the bandwidth corresponding to the spot's color. Every color may be assigned to several spots. This is the concept of spatial reuse (see for instance [5]). Observe that terminals in different spots of the same color will interfere with each other when using the same frequency band within the spots total bandwidth. Multiple terminals will not be allowed to transmit if the global interference generated is too high, as it will impair the correct reception of the data by the satellite. Color assignment is given as an entry of our problem. Examples of color assignment can be seen in Fig. 1(a), resp. Fig. 1(b), when 3 colors, resp. 4 colors, are used.

Since colors do not overlap in bandwidth, they are completely independent from each other. Hence, resource allocation can be done for each color separately. The original problem has simply to be split in the number of colors used, and each resulting problem can be solved independently from the others. Hereafter, we will consider only the problem of resource allocation within the same color. Without loss of generality, we will consider a spatial reuse of 4 colors. Let  $N$  denote the numbers of spots having the same color, and  $B$  denote the color bandwidth. We are particularly interested in the case where  $N \leq 32$ . Fig. 2 depicts the spots configuration within one color when 4 colors are used. Different spots of the same color are allowed to transmit only if the overall level of interferences is acceptable and does not impair the correct

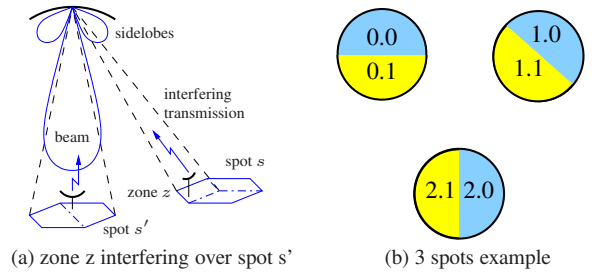


Figure 3: Interferences model and 3 spots with 2 zones example.

reception of the transmitted signals at the satellite. In the following section, we will introduce an allocation criterion as a mean to check if it is safe to activate one spot or another. This allocation criterion will condition any frequency reuse between spots of the same color.

### 2.2 Interference level

To take into account the real conditions of the radio propagation, it is necessary to account for the position of the terminals within a given spot. The spot is usually large enough to have different channel conditions in different geographical regions. We will therefore divide a spot in a number of *zones* (typically 2 or 3), assuming that each zone exhibits the same propagation conditions in all its area. The radio propagation experienced by a terminal is thus completely characterized by the zone where the terminal is.

If a terminal is transmitting at time  $t$ , using carrier  $f$ , we will say that its zone/spot is active in  $(t, f)$ . Whenever a zone is active, its transmission will generate interferences over all other spots using the same carrier at the same time. Note that this interference will be the same over any zone of a given active spot. The importance of the interference is directly affected by the size of the antennas' sidelobes. Fig. 3(a) illustrates well how a transmission can interfere over others. It is clear from Fig. 3(a) that the interference, generated over spot  $s'$  by a terminal in spot  $s$ , located in a zone other than zone  $z$ , will be different.

Let  $G(z)$  denote the minimal antenna gain corresponding to zone  $z$ . Let  $I(s, z)$  denote the maximal interference generated over spot  $s$  by a transmission in zone  $z$ . It is the maximal antenna gain in the sidelobes corresponding to zone  $z$ , when the main beam is directed to spot  $s$ . If zone  $z$  belongs to spot  $s$  then  $I(s, z) = 0$ . The received signal at the satellite is useful only if its power amplitude is large enough compared to the power of the interfering signals. In other words, the carrier to interference ratio should be beyond a certain threshold  $\sigma$ , otherwise the satellite cannot properly handle the received transmission. Hence, a zone  $z$  could be active in  $(t, f)$  if and only if the following criteria is satisfied:

$$\frac{C}{I} = \frac{G(z)}{\sum_{z' \text{ active in } (t, f)} I(\text{Spot}(z), z')} \geq \sigma, \quad (1)$$

where  $\text{Spot}(z)$  denotes the spot in which zone  $z$  is located.

Table 1: Test values of terminals types.

Type	Maximum number of time slots per frame	Maximum number of carriers per spot bandwidth
$t_1$	192	18
$t_2$	96	36
$t_3$	24	144
$t_4$	6	576

### 2.3 Interference model in numerical results

The power of the interfering signal used in (1) depends on the size of the antenna. Small sidelobes lead to weak interferences. Unfortunately, we do not have data on the power distribution of the interfering signal over all geographical areas, we will therefore assume the following: neighboring spots are the ones generating the highest interference over each other; remote spots still interfere one on each other but not as significantly. In the results of Section 5, the values in decibels of the gain  $G(z)$  (resp. interference  $I(s, z)$ ) are taken randomly in the interval  $[40, 41]$  (resp.  $[11, 15]$ ) decibels. Thus, we use these different quantities:

$$I_1(z) = \sum_{z' \text{ neighbor, active in } (t, f)} I(\text{Spot}(z), z') \quad (2)$$

$$I_2(z) = \sum_{z' \text{ active in } (t, f)} I(\text{Spot}(z), z') \quad (3)$$

$$I(z) = I_1(z) + (1 - \gamma)(I_2(z) - I_1(z))$$

where  $\gamma$  is a given weight. Equation (1) is replaced with

$$\frac{C}{I} = \frac{G(z)}{I(z)} \geq \sigma. \quad (4)$$

The interferences generated by remote spots are reduced by a factor  $1 - \gamma$ . Observe that taking  $\gamma = 0$  is equivalent to considering that all interferences are equally important (Eqs. (1) and (4) will be exactly the same), while having  $\gamma = 1$  nullifies the effect of transmissions in non-neighboring spots over the zone at hand.

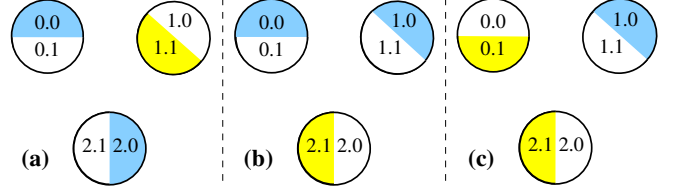
### 2.4 Types of terminals and demand

Terminals have different capabilities of transmission. A given type of terminals will use a unique frequency band. Hereafter, we will classify terminals according to their capability of transmission, and use the notation  $t_k, k = 1, \dots, \tau$  to refer to a given type of terminals. Every type of terminals  $t_k$  will be assigned a unique bandwidth, denoted by  $t_k^b$ . In our problem, the ratio of the bandwidths of any two different types is either an integer or the inverse of an integer and is called the multiplicity. Nevertheless, each type transmits the same amount of data: for any type  $t_k$ , the product of its bandwidth,  $t_k^b$ , and its slot duration, denoted by  $t_k^t$ , is a constant:  $t_k^b t_k^t = \Delta$ . Table 1 reports the values used to test our algorithm.

The individual demands of all terminals in a zone are aggregated according to the type of terminals, and hence, the bandwidth used by every type. Let  $d(z, t_k)$  denote the demand in time slots in zone  $z$  expressed in time slots of type  $t_k$ , for any zone  $z$  and any type  $t_k$ .

Table 2: Gain and interferences of the 6 zones in the example.

Zone	Gain	$I(\text{Spot } 0, \cdot)$	$I(\text{Spot } 1, \cdot)$	$I(\text{Spot } 2, \cdot)$
0.0	4	-	5	3
0.1	6	-	5	7
1.0	3	4	-	2
1.1	8	7	-	10
2.0	5	3	7	-
2.1	5	7	3	-

Figure 4: Valid 3-spot combinations for a threshold  $\sigma = 0.30$ .

## 3 A simple example

In this section, we will consider the simple case where there is only one type of terminals, i.e. all terminals use the same amount of bandwidth to transmit their data. For every carrier, the channel can be accessed simultaneously by multiple terminals/zones according to the Time-Division Multiple Access (TDMA) technique. Solving the resource allocation problem translates then into the following question: which zones are allowed to transmit in a given time slot and using a given carrier?

Consider the example illustrated in Fig. 3(b). There are 3 spots transmitting in the same color, each spot having 2 zones. When active, every zone generates a certain level of interference over all other spots (gain and interferences can be found in Table 2, values are not in dB). Every spot can have either one of its zones active, or be inactive (recall that only one zone in a given spot can be active at a given time). Hence, there are  $3^3 = 27$  possibilities in our simple example.

Considering any zone from the example, this zone can be active (*on*) only if its carrier-to-interference ratio is above a certain value. This ratio will naturally depend on whether the other spots are active or not (*on* or *off*). For every zone considered, there are 9 possible situations, as reported in Table 3. Let  $\sigma = 0.3$ . All of the situations where only two spots are active are valid, since the carrier-to-interference ratio is higher than 0.3 for all zones in every such situation (refer to last column and last row for every zone). Among all  $2^3 = 8$  situations where 3 spots are active, only 3 are valid. For instance, if zones 0.0, 1.0 and 2.0 are active, it appears that the carrier-to-interference ratio is above  $\sigma = 0.3$  for zones 0.0 and 2.0, but not for zone 1.0. The only 3 combinations with 3 active spots that are valid are illustrated in Fig. 4.

Observe that the 3-spot combinations transmit more data, at the same time, than the 2-spot combinations which are less efficient.

**Case of a simple demand** Assuming that there is a demand of 100 time slots per *zone*, it is clear that the minimum number of time slots necessary to fulfill the demand is 200, since only one



Table 3: Values of the carrier-to-interference ratio.

<i>C/I</i> for Zone 0.0	Zone 1.0 <i>on</i>	Zone 1.1 <i>on</i>	Spot 1 <i>off</i>
Zone 2.0 <i>on</i>	0.57	0.40	1.33
Zone 2.1 <i>on</i>	0.36	0.29	0.57
Spot 2 <i>off</i>	1.00	0.57	-
<i>C/I</i> for Zone 0.1	Zone 1.0 <i>on</i>	Zone 1.1 <i>on</i>	Spot 1 <i>off</i>
Zone 2.0 <i>on</i>	0.86	0.60	2.00
Zone 2.1 <i>on</i>	0.55	0.43	0.86
Spot 2 <i>off</i>	1.50	0.86	-
<i>C/I</i> for Zone 1.0	Zone 0.0 <i>on</i>	Zone 0.1 <i>on</i>	Spot 0 <i>off</i>
Zone 2.0 <i>on</i>	0.25	0.25	0.43
Zone 2.1 <i>on</i>	0.38	0.38	1.00
Spot 2 <i>off</i>	0.60	0.60	-
<i>C/I</i> for Zone 1.1	Zone 0.0 <i>on</i>	Zone 0.1 <i>on</i>	Spot 0 <i>off</i>
Zone 2.0 <i>on</i>	0.67	0.67	1.14
Zone 2.1 <i>on</i>	1.00	1.00	2.67
Spot 2 <i>off</i>	1.60	1.60	-
<i>C/I</i> for Zone 2.0	Zone 0.0 <i>on</i>	Zone 0.1 <i>on</i>	Spot 0 <i>off</i>
Zone 1.0 <i>on</i>	1.00	0.56	2.50
Zone 1.1 <i>on</i>	0.38	0.29	0.50
Spot 1 <i>off</i>	1.67	0.71	-
<i>C/I</i> for Zone 2.1	Zone 0.0 <i>on</i>	Zone 0.1 <i>on</i>	Spot 0 <i>off</i>
Zone 1.0 <i>on</i>	1.00	0.56	2.50
Zone 1.1 <i>on</i>	0.38	0.29	0.50
Spot 1 <i>off</i>	1.67	0.71	-

zone per spot can be active at any time. For the first 100 time slots, the combination in Fig. 4(a) can be used to satisfy the demand of zones 0.0, 1.1 and 2.0, and for the second 100 time slots, the combination in Fig. 4(c) can be used to satisfy the demand of zones 0.1, 1.0 and 2.1, which solves the problem.

**Case of a more complex demand** Consider here a demand slightly more complex than in the previous case, as can be seen in Table 4. The demand per *spot* is 200 time slots, as in the previous case, but more than 200 time slots are needed to satisfy all zones, because the 3 combinations of Fig. 4 cannot be used as efficiently as before. It is clear that the combination in Fig. 4(a) can still be used for 50 time slots to satisfy the demand of zone 0.0, and zones 1.1 and 2.0 are left with 100 time slots demand to satisfy. Also, the combination in Fig. 4(c) can be used for 50 time slots to sat-

Table 4: Demand of the different zones (expressed in time slots).

Zone	0.0	0.1	1.0	1.1	2.0	2.1
Demand	50	150	50	150	150	50

Table 5: A more efficient solution to the example.

Number of time slots	Family to use	Active zones
100	Zones 0.0, 1.1, 2.0	Zones 0.0, 1.1, 2.0
50	Spot 0, Zones 1.0, 2.1	Zones 0.1, 1.0, 2.1
50	Spots 0, 1	Zones 0.1, 1.1
50	Spots 0, 2	Zones 0.1, 2.0

isfy the demand of zones 1.0 and 2.1, and zone 0.1 is left with an unsatisfied demand of 100 time slots. To complete the allocation problem, we can use combinations with only two active zones, allocating 50 time slots to each one of the following combinations: (i) zones 0.1 and 1.1; (ii) zones 0.1 and 2.0; and (iii) zones 1.1 and 2.0. Observe that the allocation procedure consists mainly in allocating 250 time slots to combinations of zones, provided that these combinations are valid.

Looking at Fig. 4, we can see that combinations (b) and (c) differ only on spot 0. It is therefore possible to merge these combinations into one, composed of *any* zone of spot 0 and zones 1.0 and 2.1. Hereafter, we will use the term “family” to refer to such combination of zones/spots. Observe that it is possible to use a given family when allocating slots, even though not all zones within this family need to be active. This observation will add flexibility to the solution. Using the same amount of time slots as before, that is 250, the allocation to satisfy the demand of Table 4 could now be satisfied as expressed in Table 5. In this solution, zone 0.0 will be assigned 50 extra time slots.

## 4 Solving the general case

As seen in the previous section, to solve the allocation problem in the simple case where there is only one type of terminals, we have first computed the carrier-to-interference ratio for all zones which let us identify the valid combinations, or families, of zones that are allowed to transmit simultaneously. Second, we have allocated a certain number of time slots for some families in order to satisfy the demand of all zones. To solve the allocation problem in general (arbitrary number of zones/spots, arbitrary demand and multiple types of terminals) we will have to (i) generate families of spots/zones that are valid (see Section 4.1), (ii) identify the amount of time slots of each type to allocate to which families in order to satisfy the demand (see Sections 4.3-4.8), and (iii) allocate the required number of time slots by placing the carriers in the radio channel and the time slots in the corresponding time frames (see Section 4.2). Section 4.9 presents a wrap-up of our approach.

### 4.1 Solving interference problems

Our approach is mainly based on the following key observation: for any time  $t$  and any frequency  $f$ , there exists at least one family of zones that can be simultaneously active. Let  $Z$  denote one such family, we therefore have:

$$\frac{G(z)}{\sum_{z' \in Z} I(\text{Spot}(z), z')} \geq \sigma \quad \forall z \in Z. \quad (5)$$

Naturally, there could be in family  $Z$  no more than one (active) zone per spot. This concept of concurrent transmissions is somehow similar to graph coloring [9], where families of independent edges are used to solve the problem.

In practice, there is a very large number of families checking this criterion. It is possible to have families that differ only by one spot, according to which zone in the spot is active (see the example in Section 3). As already said, such families can be merged in

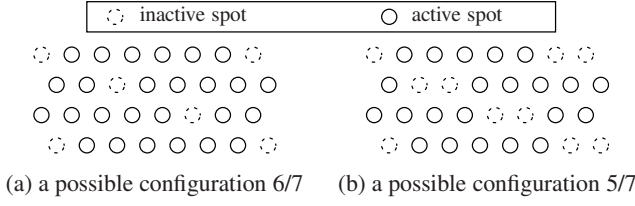


Figure 5: Example of configurations 6/7 and 5/7.

a single family. To solve the interference problem, we will generate a certain number of families, that will be used later on in the time slot allocation procedure. It is crucial to generate in the first place the most efficient families, or in other words, the families having the highest possible number of zones that can be active in  $(t, f)$ , while presenting the highest flexibility.

#### 4.1.1 Generating generic families

The threshold of interference  $\sigma$  is given as an input. If  $\sigma$  is very weak (for instance 10dB, which is not very realistic), all spots can be active in  $(t, f)$ . As  $\sigma$  increases, less spots can be active simultaneously using the same frequency. The difficulty here is to have the maximum number of active spots/zones for a given  $\sigma$ .

Recall the allocation criterion given in (4). It makes the distinction whether the interfering terminal is in a neighboring spot or not. Terminals in the vicinity are considered to interfere more than remote terminals. It then comes out that inactive spot should be geographically distributed for increased efficiency. We consider situations where only a restricted set of spots are inactive. We call a configuration 6/7 (resp. 5/7, 4/7) when at most 6 (resp. 5, 4) spots over a vicinity of 7 are active. We illustrate in Fig. 5 such possible configurations. We translate the illustrated patterns (that have maximality properties on the infinite grid) to obtain a limited but efficient series of families.

#### 4.1.2 Status of a spot

We have introduced efficient spatial configurations that can yield several families of active zones. Indeed, spots are usually divided into few zones (typically 2 or 3), and there are several possibilities for having a spot active. As (i) the power gain depends on the geographical zone within a spot, and (ii) the interferences generated over the spot depend on which zones have transmitted the interfering signals, it is quite possible that one zone in a spot does not check the allocation criterion (4) while another zone in the very same spot does. Therefore, every spot will be assigned a *status* describing which zones can potentially be active. If a spot  $s$  has  $nbZones(s)$  zones, then its status takes value in the interval  $[0, 2^{nbZones(s)} - 1]$ . For instance, the status of a 3-zone spot could take on one of the following values (a 2-zone spot could take on one of the first 4 statuses in the list):

- 0: the spot is inactive;
- 1: zone 0 checks (4), hence it could transmit;
- 2: zone 1 checks (4), hence it could transmit;

- 3: zones 0 and 1 check (4); either one could transmit;
- 4: zone 2 checks (4), hence it could transmit;
- 5: zones 0 and 2 check (4); either one could transmit;
- 6: zones 1 and 2 check (4); either one could transmit;
- 7: all zones check (4); either one could transmit;

Instead of generating families of zones, we will generate families of spots and assign to each spot the convenient status given the allocation threshold  $\sigma$ . Allocating time slots to a 3-zone spot with status 7 would actually be done by allocating the time slots to either one of its 3 zones, which increases freedom and improves the efficiency of our approach.

#### 4.1.3 Simplifying the computation of the allocation criterion

At the beginning of Section 4.1, we have defined a family of zones  $Z$  satisfying (5). In this section, we will derive a similar equation for families of spots. Instead of checking the allocation criterion (4) for every zone, we will have to check it for every spot. To be able to check if a spot could be active and decide which status it could have, we assign to every spot a gain and an interference over other spots.

The gain of a spot is defined as the minimum value of the gains of its zones which are active (information available from the status of the spot). Let  $G(s)$  denote the spot gain, we can write

$$G(s) = \min_{z \text{ in } s, \text{ active}} G(z).$$

The interference generated over spot  $s$  by spot  $s'$  is defined as the maximum value of the interferences generated by all zones of spot  $s'$  that could potentially be active. It will be denoted as  $I(s, s')$ . We have

$$I(s, s') = \max_{z' \text{ in } s', \text{ active}} I(s, z').$$

Recall the sums  $I_1(z)$  and  $I_2(z)$  introduced in (2)-(3). They represent the overall interference generated by active zones in neighboring spots and in all spots, respectively. Let  $I_1(s)$  and  $I_2(s)$  be their equivalent at the spot level:

$$I_1(s) = \sum_{s' \text{ neighbor, active}} I(s, s'), \quad I_2(s) = \sum_{s' \text{ active}} I(s, s')$$

Similarly to what we did at the zone level, the total level of interference generated over a spot  $s$  will be computed as:

$$I(s) = \gamma I_1(s) + (1 - \gamma) I_2(s)$$

Thus, a spot is said to be *valid* if it checks the following criterion

$$\frac{G(s)}{I(s)} \geq \sigma. \quad (6)$$

The advantage of using (6) rather than using (4) will be clear from the following example. Consider a spot whose status is 7. This means that it has 3 zones that could all be active (of course, not together). To check this hypothesis, one would have to check

if each zone satisfies the criterion (4). It is definitely more advantageous to use instead the criterion (6) as the computation time would be greatly reduced. Note that (6) implies (4). For any active zone  $z$  in spot  $s$ :

$$\begin{aligned} \frac{G(s)}{I(s)} &= \frac{G(s)}{\gamma I_1(s) + (1 - \gamma) I_2(s)} \\ &\leq \frac{G(z)}{\gamma I_1(z) + (1 - \gamma) I_2(z)} = \frac{G(z)}{I(z)}. \end{aligned}$$

For flexibility reasons, we would like to have all spots in a family have a status equal to  $2^{nbZones(s)} - 1$ . To that purpose, we will first generate families of spots, all having the highest status, and then test their validity. That can be done by checking the allocation criterion (6) for all spots in a family.

#### 4.1.4 Heuristics for generating valid families

We want to maximize the number of active zones, we start by generating the 7 families 6/7 in which any active spot  $s$  has the status  $2^{nbZones(s)} - 1$  while inactive ones have status 0. We then successively test the validity of these families and separate them in two pools, one for valid families and the other for non-valid families. We do the same with families 5/7, 4/7, etc.

To make a non-valid family become valid, some of its active zones should be deactivated. For instance, if a 3-zone spot having status 7 (any one of its 3 zones could be active) is not valid, then we should test the validity of its family when its status is 3, 5 or 6 (zone 2, zone 1 or zone 0 are deactivated). The following heuristic is used:

1. randomly choose a non-valid family;
2. as long as the family is not valid, do:
  - (a) randomly choose a spot,
  - (b) if its status is non-null and the spot is non-valid, deactivate at random one of the active zones; keep a record of the spot identifier;
3. try, for a certain number of times, to reactivate zones which were deactivated in step 2 and test the validity of the resulting family after each try: an amendment is adopted only if the family is valid;
4. compare the valid family obtained in step 3 with those in the pool of valid families. In case of redundancy, increment a counter of redundancies and reject the family; otherwise, add the family to the pool of valid families. Return to step 1 to generate another family.

This algorithm stops either when the desired number of valid families is reached, or when the counter of redundancies has reached a given maximum value. At this point, we have generated valid families of spots. In every spot  $s$  of a valid family,  $0, \dots, nbZones(s)$  zones are candidates in the time slot allocation procedure.

## 4.2 Placing the carriers in the radio channel

The constraints on the radio channel deal with the spot bandwidth  $B$  and the time frame length  $T$ . When planning the allocation of a time slot from a given carrier to a given type of terminal, one schematically uses a rectangle of a *fixed* surface equal to  $\Delta$  in the time-frequency space (recall Section 2.4). See for instance zone 0.1 in Fig. 7 in which two different types of terminals are used.

Thus, if the types of terminals are denoted by subscripts from 1 to  $\tau$  (ordered by decreasing bandwidth), and if  $x^{t_k}$  denotes the number of time slots of type  $t_k$  used in the spot, we then have:

$$\sum_{k \in \{1, \dots, \tau\}} x^{t_k} \leq \frac{BT}{\Delta}.$$

In other words, the maximal surface, in the time-frequency space, that can be allocated to a spot is equal to the product  $BT$ , yielding an upper bound equal to  $BT/\Delta$  on the number of time slots that can be allocated.

The following result is used to establish the properties of a filling of time slots:

**Lemma 4.1** *Let  $G = (V, E)$  be a directed graph with  $V = \{t_1, \dots, t_\tau\}$  and  $E = \{(t_j, t_k) : j < k\}$ . Define  $w_{(j,k)} = w(t_j, t_k) = \frac{t_j^b}{t_k^b} - 1$ . Then any path in  $G$  from  $t_1$  to  $t_\tau$  has a weight less than  $w_{(1,\tau)}$ .*

**Proof** Note that  $G$  is transitive. For each  $x$  and  $y$  such as  $x \geq 1$  and  $y \geq 1$ , we have  $x-1+y-1 = xy-1-(x-1)(y-1) \leq xy-1$ . Thus, if  $(t_i, t_j) \in E$  and  $(t_j, t_k) \in E$ , then  $(t_i, t_k) \in E$  and  $w_{(i,k)} \geq w_{(i,j)} + w_{(j,k)}$ . Which implies the result, by transitivity. ■

Thereafter, we show that a path in this graph corresponds to losses due to the geometrical structure of the problem. Any change in type during the placement process will incur a waste in space in the time-frequency space. Changing from type  $t_i$  to type  $t_j$  ( $j > i$ ) will cause *at most* an unused space equal to  $w_{(i,j)}$ . To minimize the space that could be lost, the best thing to do is to place the types monotonically. We have opted to fill the time-frequency space from left to right and top to bottom using the ascending order of types. The maximum number of unused time slots with this policy is given by the weight along a path in  $G$  that goes from  $t_1$  to  $t_\tau$ . We know from Lemma 4.1 that this maximum is less than  $w_{(1,\tau)}$ .

**Result 4.1** *It is feasible to place, in the time-frequency space,  $x^{t_k}$  time slots of type  $t_k$ , for  $k \in 1, \dots, \tau$  if*

$$\sum_{k \in \{1, \dots, \tau\}} x^{t_k} \leq \frac{BT}{\Delta} - w_{(1,\tau)}.$$

*This equation is therefore a sufficient condition for a placement algorithm.*

**Proof** We convey the reader to the book chapter [1] for the proof. ■

Therefore, let  $\delta = w_{(1,\tau)}$ . Observe that for the data in Table 1, this constraint allows to solve the problem of the placement by sacrificing less than  $w_{(1,4)}/3456 = 0.897\%$  of the bandwidth. It

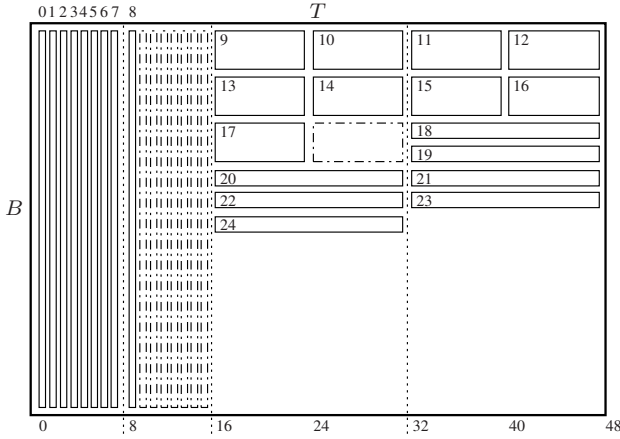


Figure 6: Sample output of the placement algorithm.

might be possible to do even better than that by adopting a lower value of  $\delta$ , assuming that the arrangement will still be feasible. In practice, one can carry out the placement according to many other policies, which may lead to a waste smaller than  $w_{(1,\tau)}$ .

Due to space limitations, we will not write here the placement algorithm. The interested reader may refer to [1] for a simple version of it. We will therefore just give an example of the final stage of an algorithm working with 3 types of terminals, as seen in Fig. 6. The rectangles drawn in dotted lines are “lost spaces” whereas the rectangles in continuous features are time slots of different types placed on the time-frequency space. The selected example being very small (demand of few time slots of the same type) and the configuration being voluntarily bad, the lost space is here very significant (8 time slots out of 33 are unused). The placement represented is based on an algorithm which fills the space from left to right and “jumps” to the order of multiplicity when there is a change in the type. The orders of multiplicity in this example are 8 between types 1 and 2, and 2 between types 2 and 3.

### 4.3 Satisfying the global demand

Instead of allocating time slots of a certain type to a spot, we propose to allocate slots to typified families, *i.e.*, simultaneously in all spots. In a typified family, distinct spots can be assigned different types. If family  $F_i$  assigns type  $t_k$  to spot  $s$ , we will note  $F_i^T(s) = t_k$ .

Initially, we will consider families with only one type. Thus, for a family  $F_i$ , we can choose a type of terminal  $t_k$  which will be used on all concerned spots (another family  $F_{i'}$  would use another type  $t_{k'}$ ). In other words,  $\forall s, F_i^T(s) = t_k$ . Such families will be denoted as 1-typified families. We place this 1-typified family, in the time-frequency space, at exactly the same place for all concerned spots, implying that all spots would use the same frequency band. In this way, we are sure that the allocation criterion is respected, because of the definition of a family. Over other frequency bands, another family could be used to satisfy another (or the same) demand.

Fig. 7 shows a possible placement of the radio resources. If  $F_i^T(s) = t_k$ , we will note  $(F_i, t_k)$  in the rectangle concerned.

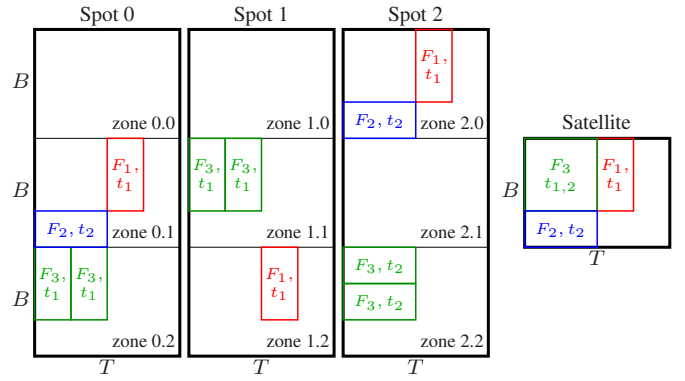


Figure 7: A global example of arranging families.

Thus, this notation is found in all active zones of a family (for instance, zones 0.1 and 2.0 for family  $F_2$ ). The constraints of capacity on each zone, in terms of bandwidth and time frame are ensured by the constraint of surface of a rectangle  $(F_i, t_k)$  on the rectangle  $B \times T$ .

A family can possibly have several types of terminals according to its different spots. It is the case, for example, for the rectangles  $(F_3, t_1)$  and  $(F_3, t_2)$ ; we will say that family  $F_3$  is 2-typified with type  $t_{1,2}$ . These families have a specific order of multiplicity. If  $t_k$  is the type in the family having the larger bandwidth and  $t_{k'}$  that with the narrower bandwidth, then the order of multiplicity of the family is  $F_i^M = \frac{t_k^b}{t_{k'}^b} \in \mathbb{N}^* - \{1\}$ .

### 4.4 Linear program

In this section, we define the linear program used to compute a solution, based on the typified families described earlier. Without loss of generality, we consider the case where each spot has three zones. We model the constraints for satisfying demands with Eqs. (8)-(10). Equation (7) provides the time-frequency space constraint of Result 4.1.

The variables of the linear program, denoted  $\mathcal{P}$ , are the  $x_{F_i}$ , which represent the number of times that the typified families are used. They must be integer variables. Let  $\mathcal{I}$  be the current set of typified families used to solve  $\mathcal{P}$ . Recall that  $d(z, t_k)$  is the demand for type  $t_k$ , as defined in Section 2.4. Let  $F_i^A(z) = on$  denote if zone  $z$  could be active, and  $F_i^A(z) = off$  otherwise.  $\mathcal{P}$  is then defined as  $\min J$  where



$$J = \sum_{i \in \mathcal{I}} F_i^M x_{F_i} \leq \frac{BT}{\Delta} - \delta \quad (7)$$

$$\forall k \in [1, \tau], \forall z \in s, \quad \sum_{i \in \Gamma(z, k)} F_i^M x_{F_i} \geq d(z, t_k) \quad (8)$$

$$\forall k \in [1, \tau], \forall z, z' \in s, \quad \sum_{i \in \Gamma(z, z', k)} F_i^M x_{F_i} \geq d(z, t_k) + d(z', t_k) \quad (9)$$

$$\forall k \in [1, \tau], \forall s, \quad \sum_{i \in \Gamma(z, z', z'', k)} F_i^M x_{F_i} \geq d(z, t_k) + d(z', t_k) + d(z'', t_k) \quad (10)$$

with:

$$\Gamma(z, k) = \{i \in \mathcal{I} / F_i^T(s) = t_k, F_i^A(z) = on\}$$

$$\Gamma(z, z', k) = \{i \in \mathcal{I} / F_i^T(s) = t_k, (F_i^A(z) = on \text{ or } F_i^A(z') = on)\}$$

$$\Gamma(z, z', z'', k) = \{i \in \mathcal{I} / F_i^T(s) = t_k, \exists z \in s / F_i^A(z) = on\}$$

It is obvious that if (7) is not satisfied, no integer solution can be found. Therefore, we choose to consider the occupied surface as the objective function to minimize. Minimizing  $J$  results in the maximization of reuse of the resources and thus in the maximization of the system throughput.

**Result 4.2** Equations (8)-(10) guarantee the satisfaction of the demand in type  $t_k$ .

**Proof** The satisfaction of the demand in type  $t_k$  can be computed on a flow from a source  $s$ , while passing by 3 arcs (or  $nbZones(s)$ , if there are  $nbZones(s)$  zones) of respective capacities  $d_0 = d(z_0, t_k)$ ,  $d_1 = d(z_1, t_k)$ , and  $d_2 = d(z_2, t_k)$ , as seen in Fig. 8. The capacities of the other arcs, denoted by  $C[z_0, z_1, z_2]$ ,  $C[z_j, z_{j'}]$  for  $j \neq j'$ ,  $\{j, j'\} \subset \{0, 1, 2\}$  and  $C[z_j]$ ,  $j \in \{0, 1, 2\}$ , are given by:

$$C[z_0, z_1, z_2] = \sum_i x_{F_i}^{t_k} \times U[F_i, \{z_0, z_1, z_2\}]$$

$$C[z_j, z_{j'}] = \sum_i x_{F_i}^{t_k} \times U[F_i, \{z_j, z_{j'}\}]$$

$$C[z_j] = \sum_i x_{F_i}^{t_k} \times U[F_i, \{z_j\}]$$

where  $U[F_i, Z]$  is equal to 1 when  $F_i$  could activate either one of the zones of the set  $Z$ , and to 0 otherwise.

The capacities of all other arcs in the figure are assumed infinite. Indeed, by the theorem of Ford Fulkerson [6] (or in its version of Menger [10]), there is a maximum integer flow from the source to the sink, which is equal to the cardinality of a minimal cut. However, there are 8 cuts of finite size (or  $2^{nbZones(s)}$  in the case of  $nbZones(s)$  zones), according to the choice of the arcs of capacity  $d_0$ ,  $d_1$  and  $d_2$ . One of these equations is trivial since it stipulates that the flow of the zones must be less than  $d_0 + d_1 + d_2$ . The 7 others are checked by our linear program. ■

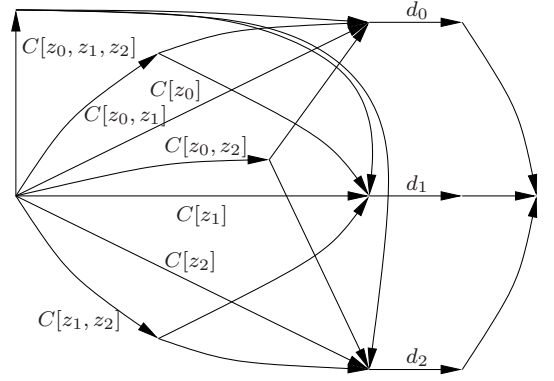


Figure 8: Modeling the constraints of zones as flows.

## 4.5 Optimal typification of families

There exists  $\tau^N$  different ways to typify a given non-typified family  $F_i$ . As it is too much to include in  $\mathcal{P}$  we will use the concept of *generation of columns* [4]. A column corresponds to one valid typified family. The optimal float solution is obtained when  $\mathcal{I}$  is the set of *all* valid typified families, a set that is too large to be used in practice. Actually, the process initializes  $\mathcal{I}$  as the set of homogeneously typified families. However, given a restricted  $\mathcal{I}$ , dual properties allow to identify new columns to be added to  $\mathcal{I}$  to improve the solution. We show in the following that dual properties characterize non-typified families, which greatly simplifies the problem of identifying an optimal  $\mathcal{I}$ .

Let  $\mathcal{P}$  be rewritten as follows:

$$\begin{aligned} \text{Minimize} \quad & f = c \cdot x \\ \text{Such that} \quad & \begin{cases} Ax = b \\ x \geq 0 \end{cases} \end{aligned}$$

Let  $A_B$  denote the matrix extracted from the corresponding system of equations, and  $x_B$  be the vector of the associated families. Let  $x_N$  denote the vector of the other families, and  $A_N$  be the corresponding matrix. In the same way, we subdivide  $c$  in  $c_B$  and  $c_N$ . We can write

$$A_B x_B + A_N x_N = b \quad \text{and} \quad f = c_B x_B + c_N x_N.$$

It comes then

$$\begin{aligned} x_B &= A_B^{-1} b - A_B^{-1} A_N x_N, \\ f &= c_B A_B^{-1} b + (c_N - c_B A_B^{-1} A_N) x_N. \end{aligned}$$

The equations above return a basic solution to the system with  $x_N = 0$ . The system is optimal if and only if

$$c_N - c_B A_B^{-1} A_N \geq 0.$$

Thus, the system is improvable if and only if a negative coefficient can be found in the above vector. We further decompose  $A_N$  by writing  $A_N = [A_{\alpha_1} \cdots A_{\alpha_j} \cdots A_{\alpha_m}]$  where  $m$  is the number of columns of  $A_N$ , each column corresponding to a family with subscript  $\alpha_j$ . In particular, we have  $c_{\alpha_j} = F_{\alpha_j}^M$ .

**Result 4.3** For any non-typified family  $\mathcal{F}$ , there exist a constant  $K_{\mathcal{F}} > 0$  and a function  $\kappa_{\mathcal{F}}$ , mapping pairs (type, Spot) to positive real numbers, such that for all typified families deriving from

$\mathcal{F}$ , we have

$$c_i - c_B A_B^{-1} A_i = F_i^M \left( K_{\mathcal{F}} - \sum_{s \text{ spot}} \kappa_{\mathcal{F}}(F_i^T(s), s) \right).$$

**Proof** Observe that, for a given line of  $A$  corresponding to a family  $F_i$ , denoted as  $A_i$ , all coefficients are either 0 or  $F_i^M$ . In addition, if  $F_i$  and  $F_j$  are typified families deriving from the same non-typified family, then  $A_i/F_i^M = A_j/F_j^M$ . Also, if  $c_i$  and  $c_j$  are the coefficients of  $c$  corresponding to  $F_i$  and  $F_j$ , then  $c_i/F_i^M = c_j/F_j^M$ . Observe that a spot  $s$  corresponds specifically to certain lines of  $A$ , given by  $P_s A$  where  $P_s$  is the corresponding projection. If  $F_i$  and  $F_j$  derive from the same non-typified family and  $F_i^T(s) = F_j^T(s)$ , then  $P_s A_i/F_i^M = P_s A_j/F_j^M$ . Last, defining the following constants  $K_{\mathcal{F}} := c_i/F_i^M$  and  $\kappa_{\mathcal{F}}(F_i^T(s), s) := c_B A_B^{-1} P_s A_i/F_i^M$  yields the result. ■ The optimal solution of our program is obtained when  $\mathcal{I}$  is the set of *all* typified valid families. Since this set is too large to be used for a computation, we simply start with a restricted  $\mathcal{I}$  which is progressively augmented to reach the optimum.

**Result 4.4** *The program  $\mathcal{P}$  with the restricted set of families  $\mathcal{I}$  is improvable with respect to the set of all valid families if there exists a non-typified family  $\mathcal{F}$  such that*

$$K_{\mathcal{F}} - \sum_{s \text{ spot}} \max_{t \text{ type}} \kappa_{\mathcal{F}}(t, s) < 0. \quad (11)$$

If we find one or several non-typified families which show that the system is improvable, we can strictly improve the solution by introducing the corresponding typified families (with the types found by the above maximization) into the linear program. This property considerably reduces the number of searches to be made in order to reach the optimal solution. In practice, as long as it is assumed that the solution is improvable, it will be possible to restrict the search by choosing a type for all spots in a subset of  $\{t_1, \dots, t_{\tau}\}$ , reducing thereby the coefficient of multiplicity of the derived families and thus, the difficulty of the integrity constraints.

## 4.6 The slave program

Given a set of non-typified valid families, the slave program assigns the types to the families and returns the exact solution of  $\mathcal{P}$  among all possible types. At first, the families are 1-typified with all possible types. The solution returns a dual which allows to derive the improving 2-typified families according to Section 4.5. Then the linear program is solved again and eventually the dual will generate new 2-typified families. The process is iterated until no new 2-typified families are obtained, which means that we have reached the optimal solution given (i) the current set of non-typified families and (ii) the fact that only 2-typified families are used. The same process is done until  $\tau$ -typified families are considered.

## 4.7 The master program

In this section, we show how we exploit the properties derived in Section 4.5 to find new valid families that will eventually lead to one  $\mathcal{I}$  having the optimal solution.

A spot  $s$  being either inactive, or either one of its  $nbZones(s)$  zones being active, it will have  $nbZones(s) + 1$  possible states. Hence, for  $N$  spots, all having the same number of zones, there will be  $(nbZones(s) + 1)^N$  combinations to test. For instance, there will be  $4^8 = 65536$  combinations to test for an 8-spot configuration in which each spot has exactly 3 zones, which is very reasonable. However, when the number of spots increases, it will no longer be reasonable to generate all families, which makes it difficult to find the optimal float solution.

Fortunately, for moderate numbers of spots, we will still be able to derive an optimal solution in a relatively small time, thanks to a pruning technique described hereafter.

- A “pruner” selects zones within a spot. If several zones have the same gain, then only one of these is selected for an exhaustive search. This step is called “pruning”.
- The  $p$  families with the highest “improvement potential” are selected. These are the ones having the highest sum in (11).
- Every selected family is “reaugmented” whenever possible. In other words, if there are zones satisfying the allocation criterion without invalidating the family, then these are incorporated in the family.

The valid families generated by this technique are added to  $\mathcal{I}$  and used in the next iteration to solve the linear program. This methodology is depicted in Fig. 9.

## 4.8 Integer solution to $\mathcal{P}$

The resolution of the slave programs enables the generation of the columns giving the best floating solution in each case. All these columns are then introduced into a new *integer* linear program, and are candidates to return the best possible *integer* solution. We stress that a solution exists with a number of non-zero variables  $x_{F_i}$  at most equal to the number of lines [3, Theorem 9.3, page 145]. For instance, in the case of 8 spots, we know that at most 224 floating variables will be used (896 in the 32 spots case), and therefore a simple ceiling of the variables will give a solution with all variables integer and multiple of 32 at less than 2.1% of the float solution (8.3% in the 32 spots case).

In practice, the resolution of the linear program, using the software Cplex CONCERT 8.0, returns an integer solution, which we arbitrarily fix at 1% of the optimal solution of the float problem. Note that solving completely the problem  $\mathcal{P}$ , using the columns candidates, cannot be achieved in a reasonable time.

## 4.9 Algorithm Wrap-up

This part sums up the whole behavior of our algorithm. Each part is represented in Fig. 10 by a rectangle (resp. an oval) corresponding to a part of the process (resp. an action or a decision). We also show the interaction between the master and the slave explained in Sections 4.7 and 4.6. The algorithm starts in the left-most rectangle. We first generate valid but non-typified families as described in Section 4.1. Then, the master program gives directly these families to the slave. The slave program operates as described in Section 4.6: the families are typified, the linear program  $\mathcal{P}$  is solved and the slave iterates until reaching optimality.

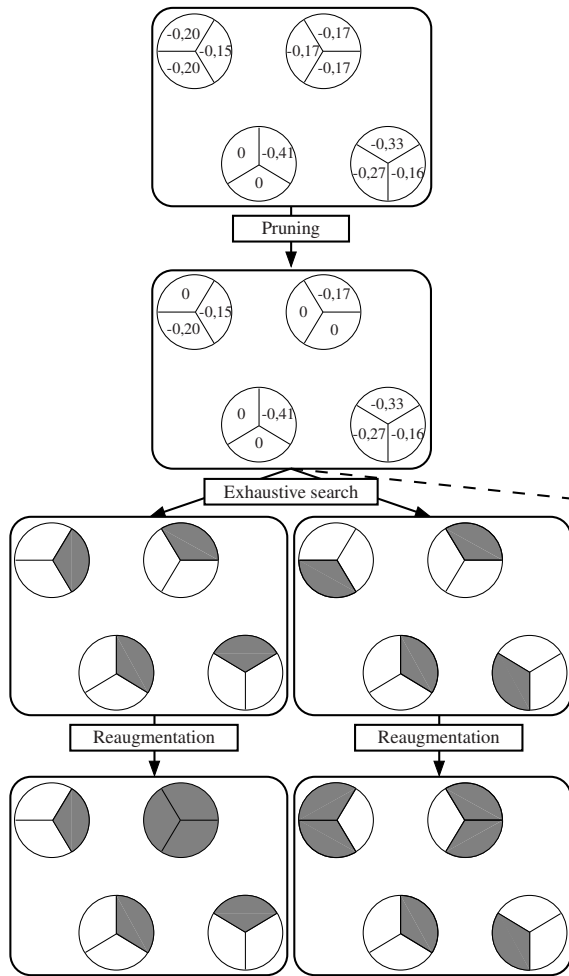


Figure 9: Pruned search of improving families.

The families involved in the solution are stored for the final integer computation. Afterwards, the master program checks the optimality of the solution given by the slave using the criterion (11). If the optimality is not reached, the pruning technique described in Section 4.7 is performed, generating new families. The master program then calls again the slave, giving it the new families generated. The master/slave process continues until optimality is reached. Next, the final integer linear program is solved as explained in Section 4.8. Finally, we achieve the placement of the resulting number of typified time slots as described in Section 4.2.

## 5 Numerical results

This section provides some numerical results returned by our approach. We have tested several configurations ranging between 8 and 32 spots. The zones demand has been generated according to examples previously provided by ALCATEL SPACE INDUSTRIES. The interferences (in dB) as well as the gains (also in dB) were drawn from uniform distributions, according to specifications provided by ALCATEL SPACE INDUSTRIES. The global interference was considered to be generated mostly by the spots in the vicinity, as the interference generated by remote spots was

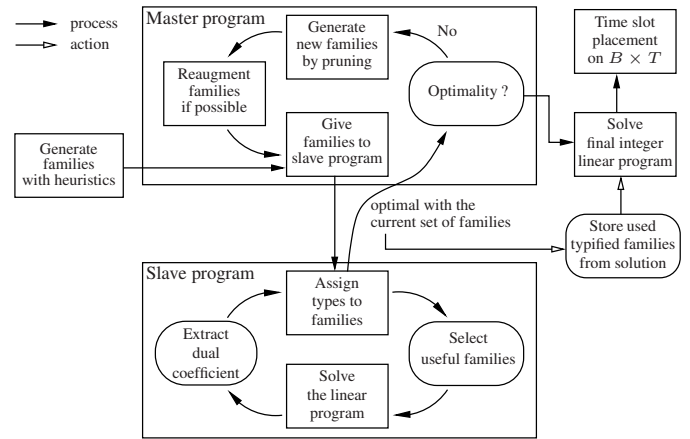


Figure 10: Algorithm overview.

reduced by 15% ( $\gamma = 0.85$ ).

Our program outputs a time-frequency plan showing the slots allocated, as it can be seen in Fig. 11. The time-frequency space therein depicted shows results in the same way as in Fig. 6. Real data, provided by ALCATEL SPACE INDUSTRIES, were used as input to our program and the results are drawn to scale. The lost space here consists of only 4 time slots.

### 5.1 Results for 8 spots

In the case where there are only 8 spots per color, our program succeeds in computing the optimal floating solution in about one minute when running on Pentium III machines. This case is particularly interesting as it enables a precise analysis of the effect of the allocation threshold.

We have computed the minimal surface, in the time-frequency plan, that is needed to satisfy the demand, for several values of the allocation threshold  $\sigma$ . The results are plotted in Fig. 12. This figure clearly highlights the fact that the minimal surface increases abruptly around certain values of the threshold. Indeed, at some point, the threshold becomes too high impairing the use of some families that will no longer be valid at the considered threshold. The “loss” of these families degrades the solution, yielding a larger minimal surface. Table 6 reports which families become no longer valid at some threshold values.

As a consequence, one is able to highlight the configurations of interferences which block the generation of good solutions. This result has obviously a very strong impact on the design of antennas.

### 5.2 Results for 32 spots

For a configuration with 32 spots, we recommend a non-optimal approach using a restricted number of families. We stick to our real-time constraints that consist in obtaining a solution in a few minutes.

Fig. 13(a) depicts the amount of time slots needed to satisfy the demand as a function of the number of valid families used, for several threshold values. Observe that when the pool of families used is larger, the required amount of time slots to satisfy

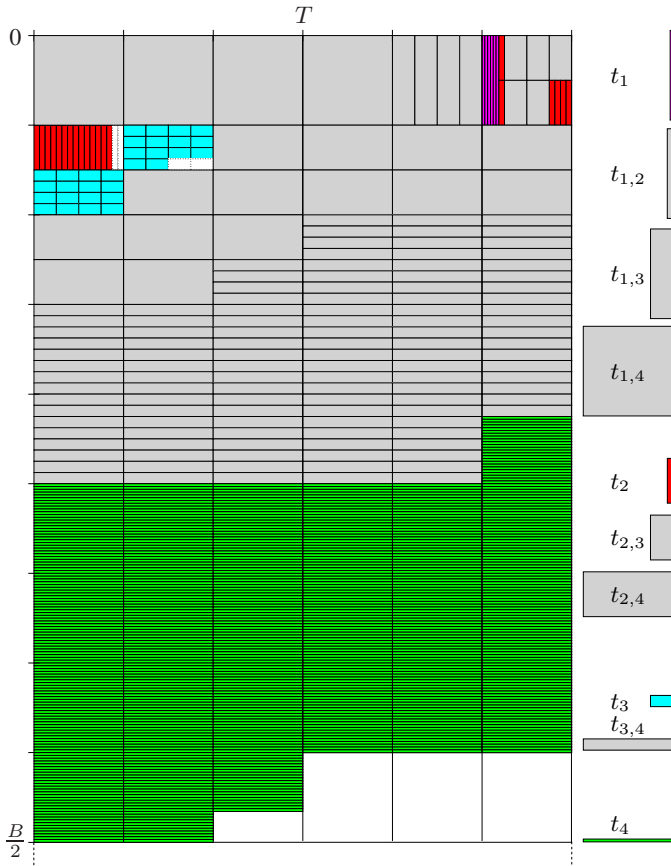


Figure 11: A sample resource allocation (satellite's point of view).

the demand gets smaller. It is therefore more efficient to use a larger pool of families. Observe as well that the solution is more efficient when the allocation threshold  $\sigma$  is smaller, regardless of the number of families used. This observation does not come as a surprise. It is obvious that smaller thresholds would allow a larger number of simultaneous transmissions. Every family would therefore include a larger number of zones that could be active, increasing the efficiency of their use.

As written previously, a larger pool of families improves the solution as it lessens the minimal amount of time slots to be allocated. However, this enhancement comes at the cost of an increased solving time, as it can be seen in Fig. 13(b). This figure plots the solving time (over Pentium III machines) as a function of the pool size, for several threshold values. Observe that, for the same number of families used, the solving time increases as the threshold values increases. This is mainly due to the time taken for generating the required amount of valid families. For larger thresholds, much more time is needed to generate valid families, as the number of non-valid families gets larger. This is why the difference, between solving times for different thresholds, increases as the number of families to generate increases (see Fig. 13(b)).

In practice, there is a trade-off between the solving time and the minimal amount of time slots to allocate. For the same number of families used, a small solving time yields a large amount

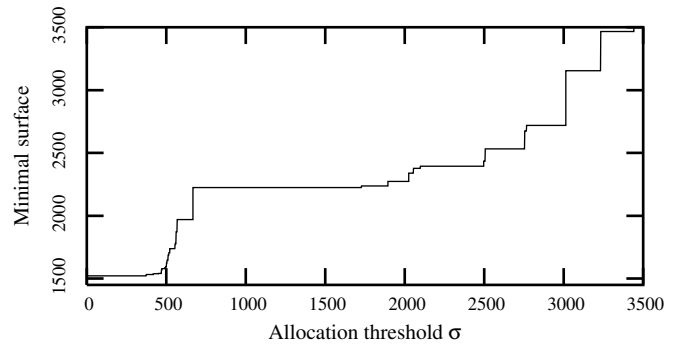


Figure 12: Minimal surface required to satisfy the demand vs.  $\sigma$ .

Table 6: Threshold values and families invalidated (X=zone off).

$\sigma$	spot 0 0 1 2	spot 1 0 1 2	spot 2 0 1 2	spot 3 0 1 2	spot 4 0 1 2	spot 5 0 1 2	spot 6 0 1 2	spot 7 0 1 2
373	X	X		X		X		
373	X	X		X		X		
418	X	X		X				
423	X	X				X		
450	X	X		X		X		
450	X	X		X		X		
450	X	X		X			X	
469	X	X		X		X		
472	X	X		X		X		
472	X	X		X			X	
472		X		X		X	X	
472		X		X		X	X	
490	X	X		X				
496		X		X		X	X	
501	X	X		X		X		
505	X	X				X	X	
510		X	X			X		

of time slots to satisfy the demand, whereas large solving times yield resource economy. It is then up to the satellite operators to decide for the optimal number of families to use, according to their priorities.

## 6 Conclusion

In this paper we have devised a novel resource allocation algorithm for MFTDMA satellites. We have considered a more accurate model for satellite communications, first by introducing a realistic modeling of the interferences that are generated by active terminals. Second, we have considered the fact that terminals have specific transmission's capabilities, which is translated into demands of different types of communications. In this context, our model is much more general than the ones found in the literature (refer to [1] for a survey).

We have first introduced the concept of non-concurrent transmissions with the use of families of spots that could transmit simultaneously at the same frequency. These families are then used to allocate time slots to multiple terminals increasing the efficiency of the algorithm. The total demand is satisfied by judi-



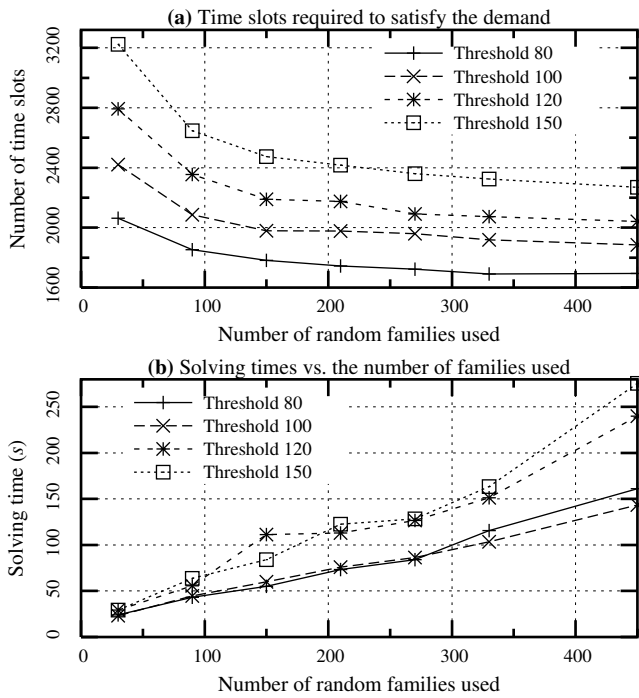


Figure 13: Results for 32 spots.

ciously placing the different carriers in the radio channel, and the time slots in the corresponding time frames. A linear program is used to compute the number of typified families to use. A column generation process improves these families and selects the good candidates for the last integer programming.

We have shown that with this solution, we can arrange the different carriers in the bandwidth with a less than 1% waste. Our numerical results for a relatively small number of spots have shown that some interference configurations are harmful, in the sense that they impair the use of some families, hence, degrading the efficiency of the solution. For a large number of spots, our results show that a large number of families can improve the efficiency of the solution at the cost of increasing the solving time. Therefore, a trade-off has to be found according to the priorities of the satellite operator.

## Acknowledgments

We would like to thank Benoît Fabre, Cécile Guiraud and Isabelle Buret, from ALCATEL SPACE INDUSTRIES, for providing many technical explanations. They have contributed to the modeling of the problem, and have shown a high interest in its resolution.

## References

[1] S. Alouf, E. Altman, J. Galtier, J.-F. Lalande, and C. Touati. Quasi-optimal resource allocation in multi-spot MFTDMA satellite networks. In M. Cheng, Y. Li, and D.-Z. Du, editors, *Combinatorial Optimization in Communication Networks*. Kluwer Academic Publishers, 2005.

[2] J. Ben-Hur. Technology Summary: Project Nemo. Gaiacomm International Corporation, <http://www.gaiacomminternational.com>, December 2003.

[3] V. Chvatal. *Linear programming*. W. H. Freeman and Company, 1983.

[4] G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.

[5] T. ElBatt and A. Ephremides. Frequency reuse impact on the optimum channel partitioning for hybrid wireless systems. In *Proceedings of IMSC '99, Ottawa, Canada*, June 1999.

[6] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, NJ, 1962.

[7] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[8] H. F. Geerdes and H. Karl. The potential of relaying in cellular networks. In *Proceedings of INOC '03, Evry/Paris, France*, pages 237–242, October 2003.

[9] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[10] K. Menger. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, pages 96–115, 1927.

[11] B. Toft. Coloring, stable sets and perfect graphs. In R. L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of combinatorics*, volume 1, chapter 4, pages 233–288. North Holland, 1995.