Proceeding of the 6th International Symposium on Artificial Intelligence and Robotics & Automation in Space: i-SAIRAS 2001, Canadian Space Agency, St-Hubert, Quebec, Canada, June 18-22, 2001.

APPROACHES FOR THE OPTIMIZATION OF THE RESOURCES MANAGEMENT IN A SATELLITE CONSTELLATION

Authors : Catherine Mancel [1][3] , cmancel@laas.fr
Pierre Lopez[3] ,Robert Valette [3], lopez@laas.fr, robert@laas.fr
Malecka Saleman[2] , Nicolas Bataille[2], malecka.saleman@cnes.fr, nicolas.bataille@cnes.fr

Address : (1) IXI, 76 rue de la colombette, 31000 Toulouse (France)
(2) CNES, 18 avenue Edouard Belin, 31401 Toulouse (France)
(3) LAAS-CNRS, 7 avenue du colonel Roche, 31077 Toulouse Cedex 4 (France)

Keywords : handover, resource management, optimization, Petri net, linear programming, constraint propagation, column generation, hybrid approach

Abstract

This paper presents the progressive approaches which have been studied in CNES for the resolution of the handover management problem. The first approach was based on the analysis with a conservative algorithm. This approach allowed us to resolve some cases but was limited. Event and activity scheduling tools have been evaluated and discarded. In a next step, a Petri net approach has been studied. If a solution exists, the Petri net modeling allows finding it but doesn't allow choosing the optimal solution within the set of potential ones. Finally, a new formalism which allows finding the optimal solution was established. It provides results in simplified configurations representative of all the constraints and was validated with a linear programming tool with little adaptation of the formalism. Then the paper will discuss the study perspectives for a hybrid global method. The principal idea is to combine the Petri net approach with an optimization method (constraint programming, graph theory, mathematical programming, combinatorial optimization …).

1. Introduction

The use of LEO (Low Earth Orbit) satellite constellations offer many advantages for the communication missions, such as easy replacement in case of satellite failure, relative robustness of the system, low cost terminals or short time transmission. In turn, it leads to management difficulties concerning the commutations of the links between the satellites when permanent transmission is required (handovers). Indeed, because of the movement of the LEO satellites and in order to guarantee the continuity of the communications to the users, it is necessary to switch over periodically the traffic from one satellite to another one. This requires having both satellites in visibility (double visibility) during the period of time required to hand over the links of the ground station and of all the users managed by this station.

The dynamic allocations of the constellation resources must be compliant with the global communication requests while using a minimal number of satellites in order to minimize the overall cost.

The objective of the optimization of the constellation resource management is to minimize the number of handovers (or maximize duration of the communications between two handovers) in order to reduce the management costs, to minimize the risk of communication interruptions.

The complexity of the problem is due to many factors, such as conflicts in the allocation between several ground stations (gateways), temporal constraints like handover duration and minimum communication duration, decision dates to start the handovers and choices of the correct resource assignments, which avoids the absence of solutions in future. Other difficulties are induced by the great number of satellites and ground stations (tens of satellites, hundreds of ground stations) and by a lot of choices for the decision date of switching. Indeed, this date can be located anywhere within the period of double visibility. All those considerations lead to a combinatorial explosion.

The different progressive approaches used in CNES to solve this kind of problems are presented hereafter.

2. Approach By Analysis

The first approach was based on the analysis with a conservative algorithm. Two handover management strategies have been considered. The first one consists in switching systematically to the satellite in visibility with the best elevation. The second one gives priority to the already established link. The first strategy ensures that the system is working in the best configuration from a link budget point of view but increases the number of handovers. The second strategy presents the interest to reduce the number of handovers. Both strategies could lead to a situation where a requested handover does not fulfill the condition of double visibility during a sufficient period while changing satellites earlier and not towards the best one (best elevation) would have allowed the eviction of this situation.

This method allows us to find optimal solutions to some problems according to the strategy considered and particularly when only one path is requested. In case of two or more paths (or several gateways closely located), the algorithm turns very complex.

3. Petri Net Based Approach

The fact that the handover constraint is a complex resource allocation policy, has led us to consider Petri nets [Mur 89]. Ordinary Petri nets are well suited to represent discrete event systems including resources. Extensions have been defined to take into account a dense time and to represent time constraints (time Petri nets, stochastic Petri nets). Other extensions allow to take into account real or integer attributes attached to the tokens (coloured Petri nets).

Representing handover constraints without the numerical constraints resulting from the visibility windows of the satellites with respect to the earth stations is easy. Let us consider the Petri net in figure 1. It represents the construction of g paths in a constellation of s satellites (assuming that one satellite can only be assigned to one gateway). Transition ti represents the first satellite assignment (beginning of the first segment of the path). Transition tf is the end of the last path segment ; the last satellite is released. Firing sequences t1→t2 describe the handovers. To connect one path segment to the next one, it is necessary to first assign the next satellite and then to release the preceding one after the handover time duration.

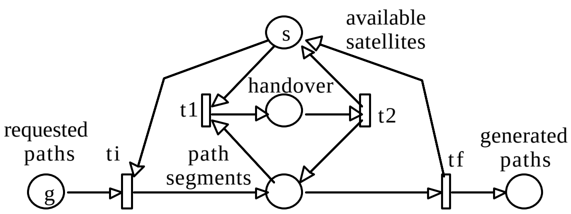

Figure 1 : Petri net model of handover constraints

A solution of the problem is then a firing sequence starting with n firings of transition ti (one assignment for each gateway), ending with g firings of transition tf and comprising a certain number of sub sequences t1→t2 (one for each handover of a gateway). Finding the optimal solution implies building all the possible sequences and choosing the best one, for example the one for which the number of handovers is minimal.

The minimal duration constraint for the handover can be represented easily, just by attaching this duration to transition t2 (time Petri net). In contrast, the fact that transitions ti and t1 can only be fired if the satellite (denoted by a token in place "available satellite") is visible from the gateway for which the path is being built (denoted by a token in place "requested paths" or "path segments") requires to attach all the necessary data to the tokens and to use good heuristics to choose good pairs of tokens (verifying the constraints and close to optimality). The management of the data attached to the tokens is cumbersome because dates are in "absolute" time. They are not a direct consequence of the duration of some activities related to places. In a similar way, the definition of the heuristics is complex and Petri nets are not very well-suited to do this. The actual Petri nets which have been simulated are in consequence large and complicated [Daf 99]. It is why a pure Petri net based approach is not convenient for this kind of problem.

Some preliminary results have however been obtained with this approach [Daf 99] by using a Petri net simulator having the capability of dealing with data structures attached to the tokens [Mis 00]. The Petri net is represented in figure 2.

Data such as the starting and finishing dates of the visibility windows, the satellite and path identifiers, the handover dates, etc., are stored as token attributes. Time is increased by small increments (module M1). In module M2, when the current time is equal to the starting date of a visibility window, a token denoting this window is added in place "visib.wind.". When the current time is equal to the end of the visibility window, transition "end visi." is fired and the corresponding token in place "visib.wind." is removed. This allows decreasing the number of data and variables which have to be simultaneously considered. The role of module M3 is just to store the series of handover for each path as token attributes. Each time a token is full, the fragment of solution is definitively stored (compression).

The role of places "wait1" and "wait2" is to synchronize the construction of all the paths with the global clock. When all the paths have been prolongated, time has to be incremented (M1) and the visibility windows updated (M2). Then all the tokens denoting paths are transferred to place "ready". This place has three output transitions. Transition "fail" is fired if the visibility window terminates and it is not possible to make a handover to prolongate the path. Transition t1 is fired if a handover can be initiated and transition "continue" is fired if there is no reason to initiate a handover.

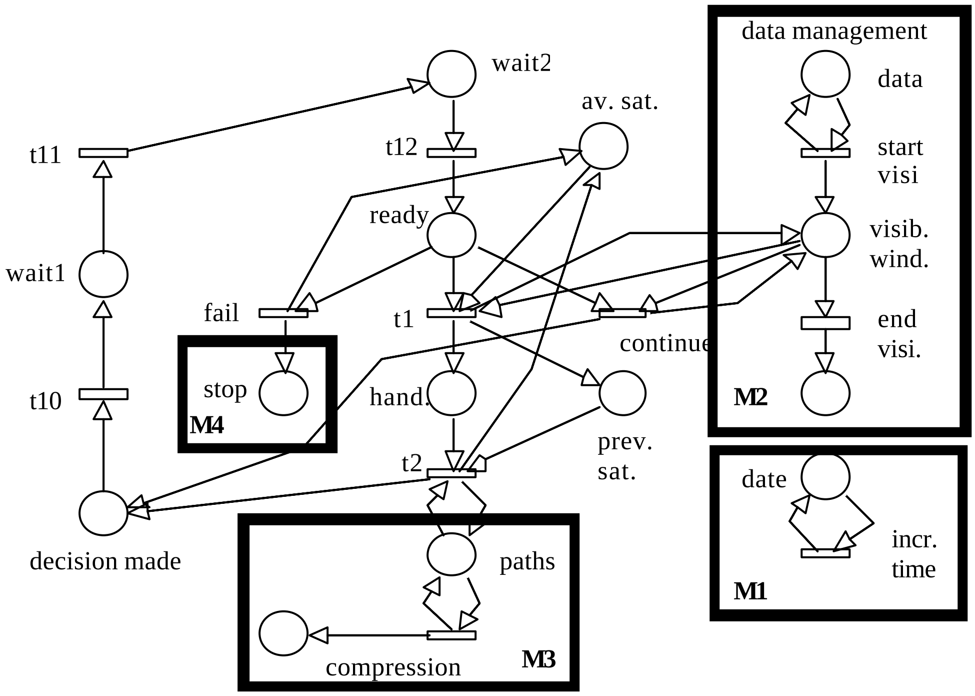

Figure 2 : Petri net used for solving the problem

We have just clearly separated the request of a new segment (place "ready") from the fact that the path has been prolongated ("decision made").
The example in figure 3 illustrates the results given by this approach. A new assignment is decided before the end of visibility of the current satellite used and taken into account the eventual need of several handovers. In figure 3, we can see that the use of satellite 2 is very short. In fact, this satellite allow the continuity of the path between satellite 1 to satellite 3 since the duration of the double visibility between satellites 1 and 3 is not sufficient to realize the handover without interruption.

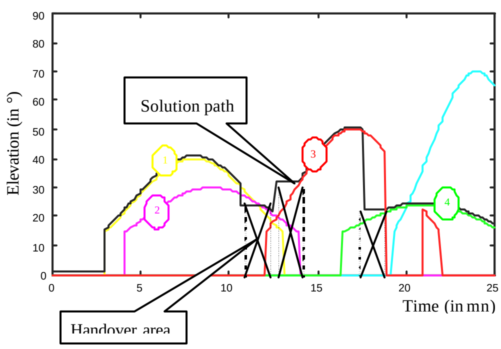

Figure 3 : Solution given by Petri net method for one gateway over 25 mn time window

This method is well-suited to manage the conflictual situations and to find solutions even it is not the optimal one. The limitation of this approach is that it is difficult to store and to retrieve all the required data when they are stored as token atributes. The approach is a greedy one (no backtracking) and to avoid situations for which it is impossible to prolongate a path, it is necessary to implement good heuritics for choosing the good satellite for each handover (when firing transition t1). It is possible to implement some heuristics in the Petri net model [Daf 99], but it is a complex and error prone task. It is the reason why a pure Petri net approach is also not satisfactory.

4. An Integer Linear Programming Formulation

4.1 Constraint identification

The aim of this approach is to give a mathematical formulation [Wol 98] to the handover management problem. This means that all the constraints presented in section 2 have to be captured by means of inequations (linear if possible, involving decision variables and parameters).
A satellite is said visible for a terrestrial user if a satisfying radioelectrical link can be established between them. For a fixed terrestrial user, each satellite of a LEO (Low Earth Orbit) constellation is only visible during some time windows, termed visibility windows. Communication links may only be established during these visibility windows. To reduce the number of assignments to search, a set of users managed by a same gateway can be assimilated to a unique user represented by the gateway. Thus, in the following, we only consider the assignments between gateways and satellites.
The required continuity of Earth/satellite communication links results in double visibility constraints. In order to hand off a communication, it is necessary to begin the following link a given time ΔH

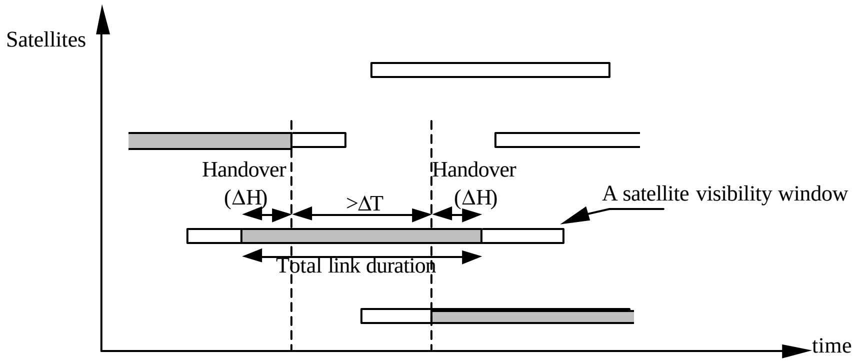

Figure 4 : The handover management problem

(about 60s) before breaking the current one. Thus, for each gateway, some visibility windows have to overlap each other during a time of at least ΔH and each assignment segment has to overlap the precedent one during ΔH exactly. Moreover, for cost and performances reasons, a gateway/satellite assignment has to contain a minimum communication duration ΔT (about 120s) out of handover periods.

Figure 4 shows the characteristics which have been just presented ; a dashed area inside a time window represents an assignment segment while a "blank" time window means none assignment has been realized in this window.

Satellite resources are constrained too. The number of instantaneous communication links with a satellite is limited because each satellite is able to communicate only on a given number of spectrums.

During a handover, two satellites are used simultaneously and the quality of radioelectrical link may be not so high as usual. Thus the best solutions to the handover management problem are the successive assignments of each gateway to satellites which allow the minimization of the number of handovers. Note that minimize the number of handovers is similar to minimize the number of assignments.

4.2 The model

To model the constraints identified in the previous paragraph we introduce the following variables and parameters :

- T : horizon of computation.
- i : assignment serial number, i =1,2, …, nsup where nsup is an upper bound of assignments number per gateway during T. Moreover a value ninf will be used in the model; it stands for a lower bound of this number.
- j : satellite number, j = 1, 2, …, s ; s is the number of satellites at hand.
- k : gateway number, k = 1, 2, …, g ; g is the number of gateways at hand.
- m : visibility window serial number between a satellite and a gateway during T ; m = 1, 2, …, w where w is an upper bound of the number of visibility windows there can exist between a satellite and a gateway during T.
- Sv_{jmk} : start time of visibility window m of satellite j for gateway k.
- Fv_{jmk} : finish time of visibility window m of satellite j for gateway k.
- X_{ijmk} : binary variable which is 1 if the assignment i of gateway k is on visibility window m of satellite j, 0 otherwise.
- S_{ik} : start time of assignment i of gateway k.
- y : upper bound of the number of assignments during T per satellite.
- z : upper bound of the number of assignments during T per satellite and per rank of assignment.

The role of the two last variables will be clarified later.

Hence, the constraints can now be written as :

$$\forall i, \forall k,\ S_{ik} \geq \sum_m \sum_j X_{ijmk}\, Sv_{jmk} \quad (1)$$

$$\forall i, \forall k,\ S_{ik} \leq \sum_m \sum_j X_{ijmk}\, (Fv_{jmk} - (2\Delta H + \Delta T)) \quad (2)$$

$$\forall i, \forall k,\ S_{i+1k} \leq \sum_m \sum_j X_{ijmk}\, (Fv_{jmk} - \Delta H) \quad (3)$$

$$\forall i < nsup,\ \forall j,\ \forall m,\ \forall k,$$

$$S_{i+1k} - S_{ik} + (1 - X_{i+1jmk})M \geq X_{i+1jmk}\, (\Delta H + \Delta T) \quad (4)$$

$$\forall i \leq ninf,\ \forall k,\ \sum_m \sum_j X_{ijmk} = 1 \quad (5)$$

$$\forall k,\ S_{1k} = 0 \quad (6)$$

$$ninf \leq i < nsup, \forall k,\ \sum_m \sum_j X_{ijmk} \geq \sum_m \sum_j X_{i+1jmk} \quad (7)$$

$$\forall k,\ \sum_{i=ninf\ to\ nsup} \sum_{m | Fv_{jmk} \geq T} \sum_j X_{ijmk} = 1 \quad (8)$$

$$\forall j,\ \sum_i \sum_m \sum_k X_{ijmk} \leq y \quad (9)$$

$$\forall j, \forall i,\ \sum_m \sum_k X_{ijmk} \leq z \quad (10)$$

Constraints (1) to (4) concern temporal aspects of the handover management problem, constraints (5) to (8) guarantee the continuity of assignments for each gateway and constraints (9) and (10) deal with the problem of the limited capacity of satellites.

Constraints (1) and (2) combine to ensure that each assignment is achieved during a visibility window. In other words, each S_{ik} has to be chosen in the corresponding window $[Sv_{jmk}\ ;\ Fv_{jmk} - (2\Delta H + \Delta T)]$. S_{ik} cannot be greater than $Fv_{jmk} - (2\Delta H + \Delta T)$ since $(2\Delta H + \Delta T)$ is the minimum duration of each assignment

(two required overlap periods ΔH to hand off the link plus a minimum communication duration ΔT). Note that the end time of assignment is not a variable of our problem. Indeed, this date is equal to the next assignment start time plus ΔH.
Constraint (3) stipulates that for each gateway, each assignment has to start at least ΔH before the end of the previous assignment visibility window.
Constraint (4) states that the minimum duration between two consecutive assignment beginnings is $\Delta H+\Delta T$. Note that in this formula M is a positive number, sufficiently large to bring the inequation to be trivial if $X_{ijmk} = 0$.
Constraint (5) and (6) ensure the initialization of the solution : for each gateway there must be at least ninf assignments and the first one has to start at 0.
Constraint (7) enforce assignments to have a logical serial number. It means for each gateway that if there is an assignment i+1 (i.e. j, m exist so that $X_{i+1jmk} = 1$) then there is necessarily an assignment i (i.e. j', m' exist so that $X_{ij'm'k} = 1$).
Constraint (8) guarantees that each gateway is assigned to a satellite at T. Thus, the continuity of assignments is ensured on the entire computation period.
It is difficult to linearly formulate the satellite resources sharing constraint. Indeed, this kind of constraint would need a great number of calculations to be tested. In order to keep our program linear and tractable we introduce two variables y and z and constraints (9) and (10). Constraint (9) means that for each satellite j the total number of assignments to j during T is bounded by y. Constraint (10) means that for each satellite j and for each assignment serial number i the total number of i^{th} assignments to j during T is bounded by z.
We expect that the minimization of y and z will allow the satellite resources sharing constraint to be respected. Thus, the precise capacity of satellites is not exactly expressed in the program, but the charge will tend to be well shared out between the different satellites if variables y and z are injected in the minimization criterion.
The solution to our problem has to minimise the number of achieved handovers during T or, which is the same, the number of assignments. Then, the objective function to minimise can be written as :
$\sum_i \sum_j \sum_m \sum_k X_{ijmk}$
which represents the total number of assignments during T. This expression has to be augmented by variables y and z, so the appropriate objective function becomes :
$\sum_i \sum_j \sum_m \sum_k X_{ijmk} + y + z$
The integer programming model provides an easy way to handle the handover problem. Nevertheless it is to be feared that the "big M" formulation (using the large number M) will not be able to deal with large scale systems.

4.3 Validation and results

In order to validate the model given in the previous paragraph a linear programming software was used. The combinatorial feature of the handover management problem is too high to allow us to test the linear program on the entire problem. Thus, the model was validated on an isolated sub-problem including only four gateways on a one hour-long horizon. Figure 5 shows the computed solution for one of these four gateways.
An optimized solution was obtained for this example. This result validates our formulation of the handover management problem constraints. However, computation duration needed to treat this sub-problem as well as the great number of generated variables and constraints shows that integer linear programming is ill-suited (at least in the form presented in Section 4.2) to solve the global handover management problem on his own.

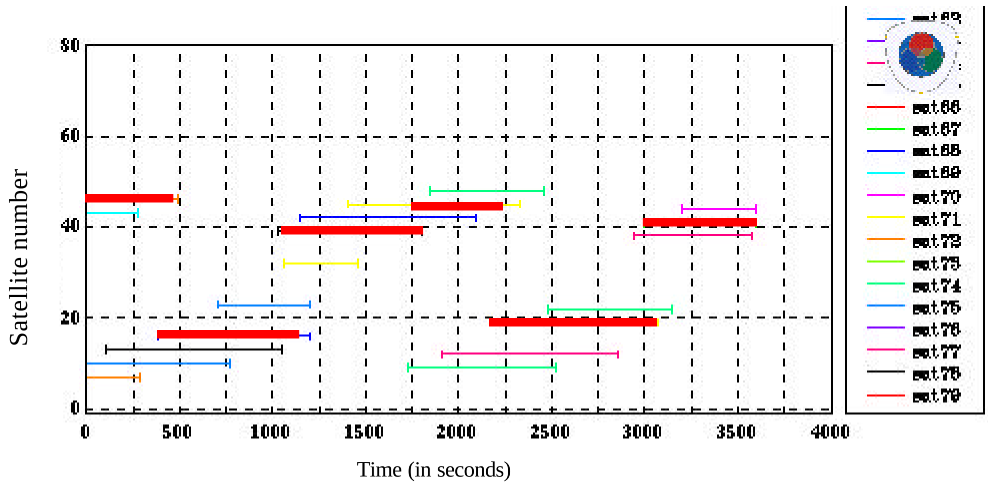

Figure 5 : Optimised Solution for one among the four gateways

5. Work in Progress

We have seen that a classical linear programming software is ill suited to efficiently solve the linear program presented in section 4.2 for the entire handover problem. Hence further works are to be investigated in order to solve the problem in a better manner. To reach this objective a way could be to design a hybrid approach in which the most recent results within Petri nets theory and integer programming will be combined with constraint propagation techniques.

5.1 Pre-processing by constraint propagation

A further work could be to envisage a constraint propagation phase in a pre-processing level. Constraint propagation involves logical processes for reducing the search space in combinatorial problems [Esq 95]. These processes are implemented by filtering procedures that mainly arise in artificial intelligence as well as in operations research, in particular when dealing with scheduling problems [Huy 00]. Consider for example the situation described in Figure 6. Since gateway 1 will necessary use satellite 2 in the interval Δ=[a–ΔH, b+ΔH], the initial set of visibility windows of this satellite for a gateway 2 can be adjusted by removing Δ to it.

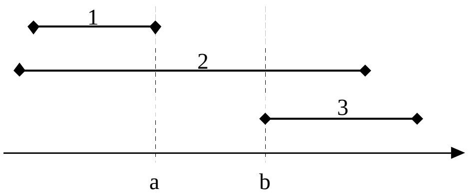

Figure 6 : Visibility windows of satellites 1, 2, and 3 for gateway 1

In the general case the initial set of visibility windows is defined by a partition of intervals ($[Sv_{j1k},Fv_{j1k}]\cup[Sv_{j2k},Fv_{j2k}]\cup\ldots$). In all cases the result of the intersection of this set with Δ is that the length of some time windows is less than two times the necessary duration to process the handovers. Such windows must be suppressed by the filtering procedure; this can lead to a drastic reduction in the number of generated variables X_{ijmk}.

The previous reasoning takes account of two gateways competing for common satellites. It must be extended to handle more general cases. In particular the processing must allow us to detect a situation where the assignment problem can be split into independent subproblems. For example in Figure 7, it is obvious that satellites from 1 to 3 must be assigned to gateways from 1 to 3; as a consequence the set of possible assignments for gateway 4 is reduced to satellites 4 and 5.

Particular cases have been studied in [Daf 99] in order to identify practical elementary conflicting situations. As an alternative to a pre-processing phase – or better, in complement – notice finally that constraint propagation mechanisms can be efficiently used at each node of a decision tree for finding good solutions in the real-size optimization problem.

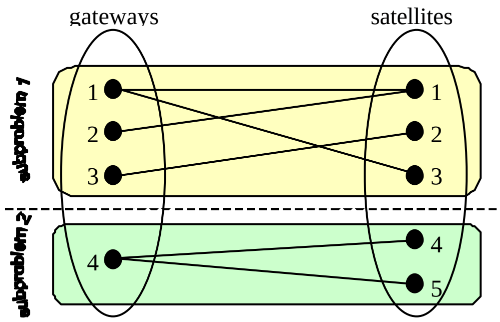

Figure 7 : Independent subproblems relating to visibility constraints

5.2 Petri nets within a hybrid approach

In order to study the possible role of a Petri net modelling within a hybrid approach, it is first necessary to point out clearly the descriptive power of a Petri net and what kind of analysis can be done in the context of optimization under a set of constraints. A Petri net is a way of describing the behaviour of a discrete event system by intention. This means that the set of all possible states (the Petri net markings) and that of all possible events (state changes associated with transition firings) are not a priori enumerated. They may be derived from the Petri net model, after defining its initial marking.

We will illustrate here how a Petri net model can be an aid to derive partial solutions. The Petri net in figure 8 represents the assignments of satellites to gateways and the fact that the satellites are only available during the visibility windows. It is a high level Petri net because attributes are attached to its tokens. Each handover corresponds to one firing of transition "assignments". A new satellite j' is assigned to gateway k and the preceding satellite j is released. This is only possible if the m'th window of j' for k is available. Transition "update visi. window" is fired each time a new visibility window (numbered m+1) is generated for a satellite j and a gateway k. It can be remarked that during the firing sequence of the Petri net a set of consistent assignments of variables X_{ijmk} (involved in the constraints (1) to (10) in section 4.2) is derived. Each time a token X_{ijmk} appears in place "path segments", the corresponding variable is set to one. The variables X_{ijmk} which have not been associated with a token are set to zero. If the firing sequence is a legal one (transitions are only fired if enabled), then the set of the obtained values for the X_{ijmk} is consistent. This means that a satellite cannot be simultaneously assigned to two gateways and that the next satellite is always assigned to a gateway before the preceding one is released.

The set of values of variables X_{ijmk} produced by a firing sequence is consistent with respect to the resource assignments. But many of these sets will be inconsistent with constraints (1) to (10) when the values of continuous parameters such as Sv_{jmk} and Fv_{jmk} are taken into account. In addition, generating all the legal firing sequences is a cumbersome task. Because of the interleaving semantics of the reachability marking graph of a Petri net all the firing sequences which only differ by the firing order of concurrent transitions produce the same set of values of X_{ijmk}.

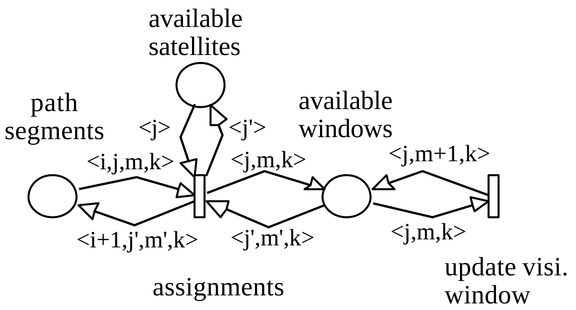

Figure 8 : assignments of satellites to gateways

Recent results about Petri nets and linear logic [Pra 99] provide a way to solve this second problem. By replacing the generation of firing sequences by the construction of proof trees in linear logic, the combinatorial explosion due to concurrency is avoided. Only one proof is done for the derivation of one consistent set of values of the X_{ijmk}. However, the set of consistent (from the point of view of the discrete constraints) values of the X_{ijmk} is still too large because the assignments of the satellites do not take into account the visibility windows (parameters Sv_{jmk} and Fv_{jmk}). This means that the hybrid approach cannot be simply a sequence of two resolutions, first the discrete constraints and then the optimal solution for the continuous ones. The main continuous constraints have to be taken into account during the construction of the proof tree at each decision node. This can be efficiently done for instance by means of an adequate constraint propagation.

5.3 Improving linear programming

A brute force use of linear programming such as presented in section 4.2 could be improved by column generation [Las 70]. This method is indeed well suited to treat large problems - with many variables (columns) or many constraints (rows) or both. The main idea is to solve a linear program without having to examine all variables.
Generally, a large linear problem has a sparse constraint matrix and it can always be formulated in a p-block angular structure, as shown in figure 9, where blocks B1 to Bp correspond to disjoint sub-problems and block B0 contains coupling constraints.

The column generation procedure consists of two phases [Leb 00]:

- First, the initial linear problem is transformed by change of variables into an equivalent "master program" with fewer constraints but many more columns (too many to be explicited).
- Then, in order to solve the master program without having to tabulate all the columns, a "column generation algorithm" is designed. This algorithm interacts with the master program and computes columns able to improve the objective function when needed.

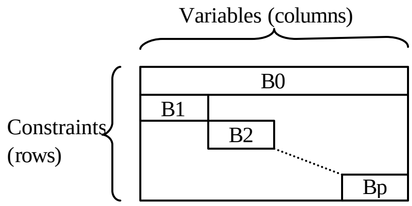

Figure 9 : The p-block angular matrix of a linear program

Figure 10 illustrates how master program and column generation algorithm interact to solve the global problem [Leb 00]. It stops when it is not possible to generate a better column.

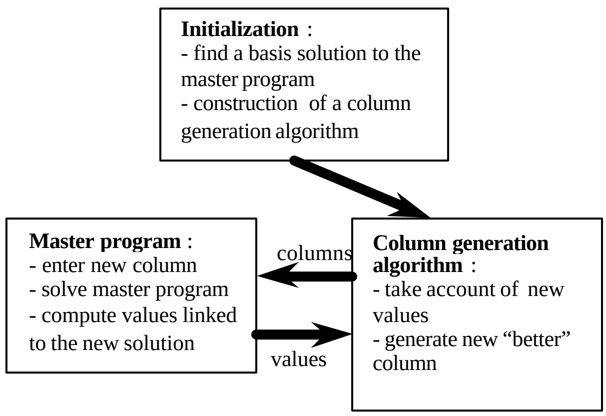

Figure 10 : Column generation procedure

Observing that both column generation and Petri net modelling are based on the decomposition of a problem in terms of its constraints, a further work could be to study possibilities to combine these methods to solve the handover problem. We could formulate the problem in a linear program with a block angular matrix as shown in figure 11.a) : block B1 represents a discrete subproblem corresponding to the part of constraints expressed in the Petri net model, block B2 the set of continuous constraints (visibility windows) and block B0 the coupling constraints. Then, we could use the proof tree construction of linear logic [Pra 99] to form the master program. This can be indeed considered as a change of variables since the proof tree generates a set of consistent sets of values of the X_{ijmk}. A column of the master program represents the successive assignments of gateways to satellites on the horizon of computation. Moreover, it allows to reducing the number of discrete constraints to one (see figure 11).

So, it seems that Petri nets and linear logic could help to design a column generation procedure for the handover problem. However, this approach does not allow to reducing the set of continuous constraints, which is very large in the handover problem.

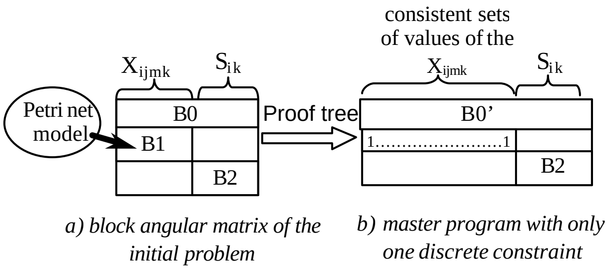

Figure 11 : Construction of a master program for the handover problem

6. Conclusion

This paper deals with the handover problem in satellite constellations. Several methods have been reviewed. Each of these methods has pointed out its advantages but also shows its limits in terms of efficacy, efficiency or modularity. Further works based on hybrid approaches such as those described in section 5 could have been investigated in order to solve this problem in a better manner. However, economic trade-off and recent technological breakthroughs have turned LEO satellite constellations less relevant for telecommunication applications. We are now addressing optimization issues in resource management for a planetary mission. A critical resource is the quantity of data which can be transferred from stations landed on the planet to the Earth via an orbiter. The arising constraints are very similar to the ones modeled in sections 1 to 5 (visibility windows for example). Hence our current research is to develop a hybrid approach based on ideas presented in section 5 well adapted to planetary mission.

References

[Daf 99]
M. Dafir : Modélisation par réseaux de Petri et optimisation des handovers dans les constellations satellitaires, mémoire de DEA, ENSEEIHT, Septembre 1999

[Esq 95]
P. Esquirol, P. Lopez, H. Fargier, T. Schiex : Constraint Programming, Belgian journal of Operations Research, Statistics and Computer Science (JORBEL), 1995, Vol.35, No.2, pp.5-3

[Huy 00]
Toàn Phan Huy : Constraint Propagation in flexible manufacturing, Lecture Notes in Economics and Mathematical Systems, Springer, 2000

[Lad 70]
L. S. Lasdon : Optimization theory for large systems, Macmillan, 1970

[Leb 00]
M. Lebbar : Résolutions de problèmes combinatoires dans l'industrie - Apport de la programmation mathématique et des techniques de décomposition, Thèse de doctorat, Ecole Centrale de Paris, Mai 2000

[Mis 00]
Miss-RdP: interactive Modelling and Simulation by Petri Nets, IXI, http://www.ixi.fr/tools/pages/miss/fr_0.htm

[Mur 89]
Tadao Murata : Petri Nets: Properties, Analysis and Applications, Proceedings of the IEEE, 1989, Vol.77, No 4, pp. 541-580

[Pra 99]
B. Pradin-Chezalviel, R. Valette, L.A. Kunzle : Scenario durations characterization of t-timed Petri nets using linear logic, PNPM'99, Zaragoza, Spain, september, 8-10, 1999

[Wol 98]
Laurence A. Wolsey : Integer programming, Wiley, 1998