

# Validation du design d'une matrice de switches dans un satellite de télécommunications

F. Bessaïh<sup>1,3</sup>, B.Cabon<sup>1</sup>, D. Feillet<sup>2</sup>, Ph.Michelon<sup>3</sup>

<sup>1</sup> EADS ASTRIUM, 31 rue des Cosmonautes, Z.I. des Palays, 31402 Toulouse Cedex 4.

{fawzi.bessaih,bertrand.cabon}@astrium.eads.net

<sup>2</sup> École Nationale Supérieure des Mines de Saint-Étienne, CMP Georges Charpak, F-13541 Gardanne, France. feillet@emse.fr

<sup>3</sup> Université d'Avignon et des Pays de Vaucluse, Laboratoire Informatique d'Avignon (EA 931), F-84911 Avignon, France. {fawzi.bessaih,philippe.michelon}@univ-avignon.fr

**Mots-Clés :** *validation de design, CSP, conflict directed back jumping, forward checking, Q-CSP, Nogoods.*

## 1 Le problème de validation de matrice de switches

La mission principale d'un satellite de télécommunications est de recevoir un signal, de l'amplifier et de le réémettre vers la terre. Après réception, le signal est démultiplexé en plusieurs signaux correspondant à autant de canaux actifs parmi ceux disponibles. Les signaux sont ensuite acheminés aux tubes amplificateurs forte puissance. Cet acheminement nécessite une matrice de sélectivité, composée de switches (voir figure 1), qui permet de sélectionner les canaux actifs à amplifier. Après quoi, ils sont acheminés vers les canaux de sortie via une seconde matrice de sélectivité, remultiplexés et réemis vers la terre.

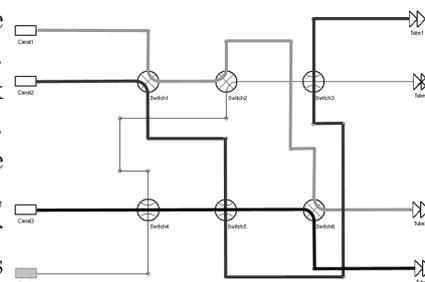


FIGURE 1 – Matrice de Switches

Les tubes d'amplifications utilisés sur ce type de satellites sont des équipements complexes et coûteux, qu'il faut pouvoir redonder de façon à assurer la mission initiale du satellite sur une longue durée (15 à 20 ans) même en cas de pannes, mais avec un nombre limité d'équipements redondants. Pour se faire, une matrice de redondance composée de switches est mise en place pour accéder à tous les tubes redondants afin de rediriger le signal d'un tube en panne vers l'un d'entre eux.

Le design de la matrice de switches (sélectivité + redondance) est validé en phase de conception de façon à envisager tous les scénarios possibles. Un scénario est une combinaison définie par un Nombre de Canaux Actifs (NCA) parmi (NC) et un Nombre de Tubes en Pannes (NTP) parmi (NT). Une combinaison est dite valide si on est capable de relier chaque canal actif à un tube actif (les couples canal-tube devant être disjoint - voir figure 1). Valider ce design revient alors à énumérer et à vérifier la validité de toutes les combinaisons, qui pour un satellite standard se compte en plusieurs millions voir milliards.

Dans cet exposé, nous allons introduire la problématique de validation de matrice de switches. Nous présenterons la modélisation, l'approche de résolution et en conclusion, les résultats et les perspectives envisagées.

## 2 Modélisation et approche

La problématique a été modélisée sous forme d'un CSP binaire, où les variables sont les canaux actifs d'une combinaison donnée (correspondant à un problème élémentaire). À chaque variable est associé un domaine composé de tous les chemins possibles reliant un canal à tous les tubes atteignables en un nombre limité de switches traversés. En moyenne, le domaine d'une variable comprend une cinquantaine de valeurs (de chemins). Les contraintes entre les valeurs sont binaires et portent sur la compatibilité entre les chemins.

Pour résoudre chaque problème élémentaire, on utilise une recherche systématique de type retour arrière Conflict Directed Back Jumping [7]. Auquel est associé :

- une technique de filtrage : l'arc consistance [5],
- une technique de propagation de contraintes : le Forward Checking [2],
- un ordonnancement dynamique des variables [3].

## 3 Résultats et perspectives

L'approche choisie s'avère être très efficace pour résoudre une combinaison, en moyenne en  $10^{-4}$  seconde lors de tests réalisés sur de véritables matrices comprenant plusieurs millions, voir plusieurs milliards de combinaisons. L'inconvénient d'une telle approche réside dans la nécessité d'énumérer et de résoudre toutes les combinaisons, ce qui pose le problème de leur nombre quand celui-ci se compte en plusieurs milliards. Par exemple, pour 18 canaux actifs parmi 27 et 4 tubes en panne parmi 24, 5 milliards ( $= C_{27}^{18} \cdot C_{27}^4$ ) de combinaisons doivent être résolues. Cet exemple est validé en 6 jours et 10 heures ( $10^{-5}$  seconde par combinaison), en parallèle sur 4 ordinateurs double cœurs 2Ghz.

La dernière génération de satellite de télécommunications est composée d'une soixantaine de tubes amplificateurs et une centaine de canaux. Il devient alors vital de s'affranchir de cette combinatoire. Dans cette optique, des tentatives de réduction de la combinatoire ont été entreprises sans succès, tels que les Nogoods[6]. Cela consiste à trouver la plus petite sous combinaison non valide (un Nogood) qui permet de conclure sur l'infaisabilité d'un ensemble de combinaisons incluant le Nogood sans devoir les résoudre. Il a aussi été question d'utiliser la solution de la combinaison C pour résoudre la combinaison C+1.

Les prochaines étapes seront consacrées à la recherche d'une méthode de validation globale. Elle permettrait de valider une matrice de switches sans avoir à énumérer la totalité des combinaisons. Deux premières pistes sont à l'étude, les Quantified-CSPs [4] et la théorie des graphes [1].

## Références

- [1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows : Theory, Algorithms, and Applications*. Prentice Hall, New Jersey, mars 1993.
- [2] Fahiem Bacchus and Adam J. Grove. On the forward checking algorithm. pages 292–308, 1995.
- [3] Fahiem Bacchus and Paul van Run. Dynamic variable ordering in csps. pages 258–275, 1995.
- [4] Marco Benedetti, Arnaud Lallouet, and Jérémie Vautard. Qcsp made practical by virtue of restricted quantification. pages 38–43, 2007.
- [5] R. Dechter. *Constraint Processing*. Morgan Kaufmann Publishers Inc., 2003.
- [6] George Katsirelos and Fahiem Bacchus. Generalized nogoods in csps. pages 390–396, 2005.
- [7] P. Prosser. Hybrid algorithms for the constraint satisfaction problem. *Computational Intelligence, Volume 9, Number 3*, pages 268–299, 1993.