

# Parallélisation de l'ordonnancement de requêtes d'observations pour une constellation de satellites

Samuel Squillaci, Stéphanie Roussel, Cédric Pralet

ONERA/DTIS, Université de Toulouse, F-31055 Toulouse, France  
{samuel.squillaci,stephanie.roussel,cedric.pralet}@onera.fr

**Mots-clés** : *satellites, ordonnancement, glouton hiérarchique itéré, parallélisation*

## 1 Problématique

Nous considérons un ensemble de requêtes d'observation de la Terre à réaliser à l'aide d'une constellation de satellites en orbite basse. Chaque requête nécessite la réalisation de plusieurs prises de vue (observations) ainsi que le téléchargement de chacune d'elles vers des stations sol. Le problème est surcontraint dans le sens où il y a généralement plus de requêtes candidates que nous pouvons en satisfaire. Une solution consiste en, pour chaque satellite, une séquence d'activités (observations ou téléchargements) à réaliser dans des fenêtres temporelles prédéfinies. Le problème obtenu s'apparente à un *Team Orienteering Problem With Time Windows* [1, 2, 3].

## 2 Modélisation du problème

Chaque activité élémentaire possède une fenêtre de réalisation autorisée qui dépend des orbites des satellites. Entre deux activités, nous définissons une durée de manœuvre. A chaque instant, un satellite donné peut soit effectuer une manœuvre, soit télécharger des données, soit réaliser une unique observation. Nous traitons de plus des requêtes *hétérogènes*, avec par exemple des requêtes *one-shot* monoscopiques et stéréoscopiques, qui nécessitent respectivement une unique photo de la zone à observer et deux photos par le même satellite, des requêtes périodiques nécessitant la réalisation répétitive d'observations sur une zone cible, ou encore des requêtes systématiques qui visent à observer la zone chaque fois que cela est possible.

Chaque requête nécessite donc potentiellement une combinaison de plusieurs observations pour être satisfaite, combinaison à choisir parmi un ensemble de combinaisons possibles. Nous avons de ce fait défini des *modes de réalisation* d'une requête, chaque mode décrivant des observations élémentaires à réaliser et des créneaux de visibilité station dans lesquels les transmettre. Chaque mode est accompagné d'une priorité ou d'un score qui décrit la qualité de réalisation de la requête (score fonction des conditions météo pour chaque observation, du respect de la période pour les requêtes périodiques...). Une solution au problème est alors définie par un ensemble de modes de réalisations choisis pour les requêtes et par le séquençement, sur chaque satellite, des activités à réaliser découlant de ces choix de modes.

## 3 Glouton hiérarchique itératif avec parallélisation

L'algorithme proposé est un glouton hiérarchique itératif. A la première itération, il tente de réaliser chaque requête avec son meilleur mode. Pour cela, il essaie d'insérer à chaque étape une requête (avec son mode choisi) dont le score est le plus haut parmi les requêtes dont l'insertion n'a pas encore été tentée pour l'itération en cours. A l'issue d'une itération, pour les requêtes dont l'insertion a échoué, on considère le deuxième meilleur mode, et une nouvelle itération

à partir d'un plan vide est réalisée. L'algorithme s'arrête si un temps de calcul maximum est atteint, si toutes les requêtes sont satisfaites, ou si les modes candidats sont épuisés.

Afin d'accélérer l'algorithme, nous parallélisons les tentatives d'insertions effectuées au cours d'une itération. Pour cela, à partir d'un majorant sur le temps de manoeuvre entre deux activités et des fenêtres temporelles autorisées pour les activités, nous détectons tout d'abord des indépendances temporelles entre paires d'activités candidates. Nous définissons alors des *composantes connexes d'activités* (CCA), qui correspondent à des groupes d'activités telles que les raisonnements réalisés sur une CCA sont temporellement indépendants des raisonnements réalisés sur une autre CCA. Cela permet, pour une requête donnée, de travailler sur plusieurs CCA en parallèle. En analysant si deux modes possèdent au moins une activité dans une même CCA, nous définissons également des *composantes connexes de modes* (CCM). L'analyse de ces CCM permet ensuite de travailler sur plusieurs requêtes en parallèle (voir figure 1).

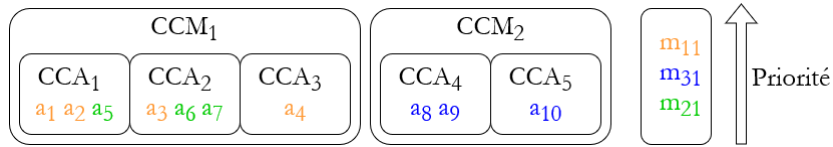


FIG. 1 – Composantes connexes d'activités (CCA) et de modes (CCM) ;  $a_i$  représente l'activité  $i$  (observation ou téléchargement),  $m_{jk}$  représentent la  $k^{\text{ème}}$  manière de couvrir la requête  $j$ .

## 4 Implémentation et expérimentations

Nous avons opté pour une architecture composée d'un processus maître distribuant des tâches à des processus esclaves. Nous recalculons à chaque itération les nouvelles CCA et CCM induites par les nouveaux choix de modes. Les meilleurs modes restants de chaque CCM sont alors candidats pour l'insertion, et les calculs associés sont parallélisés sur les esclaves qui gèrent chacun une CCA à la fois. La figure 2 donne des résultats obtenus pour des instances impliquant 32 satellites et des centaines de requêtes de différents types. Pour chaque itération, le temps de calcul va d'une seconde à environ une minute suivant l'instance considérée, et des accélérations d'un ordre de grandeur (facteur 10 par exemple) ont pu être observées par rapport à une implémentation sans parallélisation. Des expérimentations plus exhaustives restent cependant à effectuer. Ces travaux sont réalisés avec le support du Programme d'Investissements d'Avenir (projet BPI PSPC LiChIE mené par AIRBUS Defence and Space).

$\mathcal{R}^{\text{mono}}$	$\mathcal{R}^{\text{stéréo}}$	$\mathcal{R}^{\text{périodiques}}$	$\mathcal{R}^{\text{systématiques}}$	#CCA	#CCM	#A
500	0	0	0	292	80	688
250	250	0	0	280	75	934
125	125	125	125	329	1	13483
0	0	250	250	310	1	24485

FIG. 2 – Nombre de composantes connexes et nombre d'activités pour quelques instances

## Références

- [1] N. Bianchessi and G. Righini. Planning and scheduling algorithms for the COSMO-SkyMed constellation. *Aerospace Science and Technology*, 12(7) :535–544, 2008.
- [2] L. He, X. Liu, G. Laporte, Y. Chen, and Y. Chen. An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling. *Computers & Operations Research*, 100 :12–25, 2018.
- [3] X. Wang, G. Wu, L. Xing, and W. Pedrycz. Agile earth observation satellite scheduling over 20 years : formulations, methods and future directions. *IEEE Systems Journal*, pages 1–12, 2020.