

# Hybridation optimisation-apprentissage pour la planification d'observations par un satellite

Romain Barrault<sup>1</sup>, Cédric Pralet<sup>1</sup>, Gauthier Picard<sup>1</sup>, Eric Sawyer<sup>2</sup>

<sup>1</sup> DTIS, ONERA, Université de Toulouse

{romain.barrault,cedric.pralet,gauthier.picard}@onera.fr

<sup>2</sup> CNES, Toulouse

eric.sawyer@cnes.fr

**Mots-clés :** *IA hybride, réseaux de neurones, Large Neighbourhood Search, ordonnancement, satellites d'observation*

## 1 Contexte et formalisation du problème

Les enjeux liés aux missions d'observation de la Terre depuis l'espace sont multiples (gestion des risques, cartographie, défense...). Ces missions sont réalisées par des *EOS* (*Earth Observation Satellites*), des satellites en orbite basse effectuant des prises de vue d'endroits précis sur Terre. S'agissant de systèmes coûteux et complexes, il est naturel de chercher à optimiser leur usage en respectant notamment deux contraintes opérationnelles que sont premièrement les durées de transition nécessaires à la réorientation du satellite entre deux observations successives (ces durées étant *time-dependent*), et deuxièmement les fenêtres temporelles de visibilité pour chacune des zones d'intérêt. Le rôle d'un système de gestion de mission est alors de sélectionner des observations parmi l'ensemble des observations candidates et d'optimiser la séquence des observations datées à réaliser par le satellite, tout en maximisant un critère de qualité fonction de la récompense associée à chaque observation candidate.

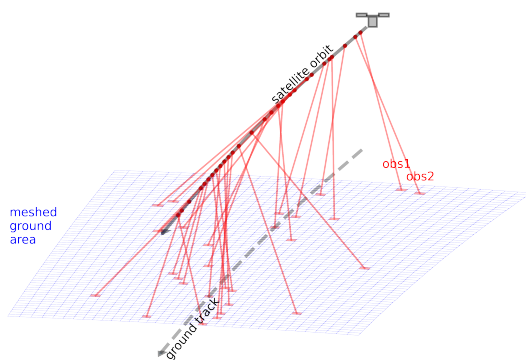


FIG. 1 – Portion d'orbite d'un satellite et mailles candidates pour une observation

Il est aisé de vérifier la faisabilité d'une séquence d'observations ordonnée, mais lorsqu'aucun ordre n'est défini a priori, cela nécessite de chercher un ordre faisable parmi tous les ordres possibles. Cela devient vite gourmand en temps de calcul, c'est pourquoi nous proposons une approche hybride où (i) lors d'une phase de sélection d'observations, la faisabilité d'un ensemble de requêtes est vérifiée par un classificateur binaire appris hors-ligne, et (ii) lors d'une phase d'ordonnancement, une séquence d'observations effectivement faisable est calculée.

Dans nos travaux sur le sujet, nous considérons un EOS sur une courte portion d'orbite contenant  $n$  requêtes d'observations sur des points d'intérêt répartis dans la zone survolée par le satellite ( $n = 200$  dans des scénarios typiques). Cette zone est modélisée sous la forme d'une grille de taille  $33 \times 65$  où chaque maille est associée à au plus une requête d'observation (cf. figure 1). On note  $\mathcal{M}$  l'ensemble des coordonnées  $(i, j)$  des mailles correspondant à une requête et on associe une récompense  $R_{i,j} \in [0.1, 1]$  à chacune de ces mailles. On cherche alors à maximiser la somme des récompenses récoltées par les observations des mailles tout en respectant la faisabilité de l'ensemble des observations au regard des contraintes

## 2 Méthode hybride de résolution

**Sélection de requêtes** La première partie de notre approche consiste donc à déterminer un ensemble de requêtes faisable par l’EOS et maximisant la somme des récompenses collectées. Cela passe par un algorithme qui, à tout instant, vérifie la faisabilité d’une sélection de requêtes de manière approchée via une fonction `feasibilityNN(·)` qui prend en entrée les valeurs choisies pour des variables de sélection  $(x_{i,j})_{i,j \in \mathcal{M}} \in \{0, 1\}$  ainsi que d’autres caractéristiques (*features*) calculées à l’aide des valeurs des  $x_{i,j}$ . Parmi ces caractéristiques figurent le nombre de requêtes dans plusieurs sous-rectangles de  $\mathcal{M}$  et leur dispersion latérale, ainsi que le poids d’un arbre couvrant minimal sur l’ensemble des requêtes sélectionnées. Comme dans une approche de type *empirical decision model learning* [2], nous avons choisi de modéliser la fonction `feasibilityNN` via un réseau de neurones entraîné hors ligne à partir d’un ensemble de données réalistes couvrant plusieurs catégories d’instances (plus ou moins denses en requêtes). Ces données ont été générées de sorte à être difficiles à classifier. La phase de test a rapporté un taux d’erreur moyen aux alentours de 11%. Hors ligne, nous apprenons également des contraintes dures portant sur le nombre maximum de requêtes qu’il est possible de satisfaire dans des sous-blocs rectangulaires de la grille. Ces contraintes de capacité offrent des conditions nécessaires de faisabilité que les variables de sélection  $x_{i,j}$  doivent vérifier à tout instant.

**Ordonnancement des observations** Le problème d’ordonnancement des observations est résolu via une méthode de type LNS (*Large Neighborhood Search*) [3]. Cette dernière génère un plan initial selon une heuristique gloutonne puis effectue des destructions et réparations du plan courant pendant une durée fixée. Notre algorithme, appelé **HySSEO** pour *Hybrid Selection and Scheduling for Earth Observation*, applique cette méthode LNS en donnant la priorité aux observations sélectionnées lors de la première phase (cf. figure 2). L’algorithme de référence auquel nous nous comparons correspond à la méthode LNS qui ne bénéficie pas de cette pré-sélection d’observations. Pour une même allocation de temps de calcul de 5 minutes, notre algorithme bat la LNS de référence dans 64% des cas et présente un *gap* d’optimalité moyen de 1.9% contre 3.4% pour sa concurrente (notre référence d’optimalité a été établie avec un algorithme LNS auquel 1h de temps de calcul a été allouée). Dans les cas où notre algorithme est battu, ce *gap* moyen vaut 2.5% contre 1.2% pour la LNS de référence. Aussi, dans la phase de sélection de notre algorithme, 28% des rejets de requêtes sont dus à la non satisfaction des contraintes dures. Ainsi, notre approche hybride mêlant réseau de neurones et apprentissage de contraintes dures a permis de battre une approche de référence pour ce problème. Une prochaine étape consiste à optimiser les plans dans un contexte multi-satellites [1].

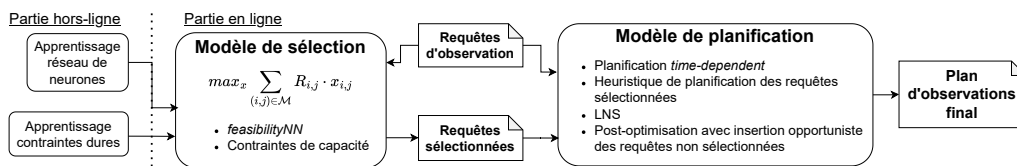


FIG. 2 – Algorithme HySSEO

## Références

- [1] Jason T. Lam, François Rivest, and Jean Berger. Deep reinforcement learning for multi-satellite collection scheduling. In *Theory and Practice of Natural Computing*, pages 184–196, 2019.
- [2] Michele Lombardi, Michela Milano, and Andrea Bartolini. Empirical decision model learning. *Artificial Intelligence*, 244, 01 2016.
- [3] Verena Schmid and Jan Fabian Ehmke. An effective large neighborhood search for the team orienteering problem with time windows. In *Computational Logistics*, pages 3–18, 2017.