Optimisation des dates d'observation de sites par un satellite

Romain Barrault, Cédric Pralet, Gauthier Picard

ONERA/DTIS, Université de Toulouse

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

Mots-clés : *profits et transitions time-dependent, programmation linéaire, programmation dynamique, méthodes heuristiques, satellites d'observation*

1 Contexte et formalisation du problème

Stimulée par des besoins en rapport avec des programmes gouvernementaux ou des initiatives privées, la recherche concernant la conception et la gestion de constellations de satellites d'observation de la Terre est en plein essor. Ces satellites artificiels en orbite basse sont utilisés pour effectuer des prises de vue, depuis l'espace, de lieux précis (appelés *sites*) sur Terre pour des usagers qui en auraient fait la demande. Ces systèmes étant coûteux et complexes, il convient d'optimiser leur usage en prenant en compte diverses contraintes. Ici, nous considérons le problème suivant : un satellite dispose d'une liste ordonnée d'observations, et il faut choisir la date précise de réalisation de chaque observation sachant que le satellite doit respecter un temps de transition entre deux observations successives afin de réorienter son instrument et que la qualité d'une observation dépend de l'angle de la prise de vue par rapport au site visé.

Plus formellement, on considère en entrée du problème : un ensemble de sites $N = \{1, 2...n\}$ numérotés de 1 à n, à observer dans cet ordre ; $\forall i \in N$, une fenêtre temporelle $[a_i, b_i]$ (noté TW pour *Time Window*) utilisable pour observer (ou "visiter") le site i (fenêtre au cours de laquelle l'angle d'observation du site i est inférieur à un seuil α_{max}) ; $\forall i \in [\![1; n-1]\!]$, une fonction de temps de transition *time-dependent* entre les sites i et $i+1$ notée $\Delta_i : t \to \Delta_i(t)$; $\forall i \in N$, une fonction de récompense *time-dependent* de visite du site i notée $r_i : t \to r_i(t)$ (fonction qui dépend de manière quadratique de l'angle d'incidence $\alpha_i(t)$ obtenu par rapport au site i en réalisant l'observation à la date t). Les fonctions Δ_i et r_i sont non triviales dans le sens où elles font appel à des librairies de calcul du domaine spatial et où on ne dispose pas de formule simple donnant directement leur valeur à toute date. Par ailleurs, pour un site $i \in N$, on dispose de la meilleure date d'observation de ce site dans $[a_i; b_i]$, notée t_i^{prox}. En pratique, cette grandeur aide à définir la fonction de récompense r_i, qui vaut alors 1 en t_i^{prox} et 0.1 en a_i et b_i. L'objectif est alors de maximiser globalement la qualité des observations effectuées. Par rapport à la littérature, le problème obtenu est à mettre en perspective avec les problèmes de tournées de véhicules avec des profits qui dépendent du temps [1, 2, 3].

2 Méthodes de résolution

Nous présentons trois méthodes de résolution : l'une utilisant de manière itérative un solveur MILP, une autre de type programmation dynamique, et enfin une méthode heuristique.

MILP itératif. Cette méthode, basée sur la programmation linéaire en nombres mixtes, réitère les étapes suivantes jusqu'au respect de précisions souhaitées :

1. **Discrétisation** des fonctions de récompense et de temps de transition sur un nombre très réduit d'instants et approximation de ces fonctions par une interpolation linéaire ;
2. **Résolution** d'un modèle MILP permettant d'aboutir aux instants optimaux pour les prises de vue (optimalité en termes de somme des récompenses collectées) ;

3. **Évaluation** des différences entre d'une part les valeurs des temps de transition et récompenses obtenues via les interpolations linéaires, et d'autre part les valeurs fournies par les fonctions de départ Δ_i et r_i. Si une de ces différences est supérieure à une précision fixée initialement, on raffine la discrétisation en y ajoutant les instants problématiques.

Programmation dynamique. Cette méthode repose sur la discrétisation du temps à partir d'un pas de temps T. Pour tout $i \in N$ et toute valeur $t \in TW_i = [a_i; b_i] \bigcap (T\mathbb{Z})$, on définit $r_{i,t} = r_i(t)$. On calcule ensuite par programmation dynamique des termes $R_{i,t}$ qui donnent la meilleure récompense d'un plan visitant les sites 1 à i et visitant le site i à la date t. Les termes $R_{i,t}$ sont initialisés par $R_{1,t} = r_{1,t}, \forall t \in TW_1$, puis on applique, pour tout site $i \in N \backslash \{1\}$ et toute date $t \in TW_i$, l'équation de programmation dynamique $R_{i,t} = r_{i,t} + \max_{t' \in TW_{i-1}, t' + \Delta_{i-1}(t') \leq t} R_{i-1,t'}$. La récompense totale de la solution correspond alors à $\max_{t \in TW_n} R_{n,t}$ et il est possible d'extraire des dates d'observation optimales simplement sur la base des argmax mémorisés à chaque étape.

Réduction itérative des fenêtres temporelles. Cette méthode fait appel à une fonction qui est capable d'indiquer très rapidement s'il existe une solution faisable étant donné des fenêtres temporelles autorisées pour observer les sites. De plus, pour tout $i \in N$, la fenêtre initiale TW_i est découpée équitablement en un nombre n_{windows} de sous-fenêtres centrées en t_i^{prox}, et on attribue à chaque sous-fenêtre une récompense égale à la valeur de r_i à la date de début de la sous-fenêtre. À partir de là, tant qu'il est possible de restreindre des fenêtres, on choisit à chaque itération de l'algorithme le site dont la fenêtre actuelle est de récompense minimale et on la restreint d'un cran à condition que le problème reste faisable après restriction (d'où l'utilité de la fonction rapide mentionnée précédemment).

3 Résultats et perspectives

Ces trois méthodes ont été testées sur 5 instances impliquant $n = 20$ sites à observer, et les résultats de ces tests préliminaires sont consignés dans le tableau ci-dessous.

	MILP itératif		Programmation dynamique		Réduction itérative des TW	
	Cpu time (ms)	Récompense	Cpu time (ms)	Récompense	Cpu time (ms)	Récompense
Instance 1	194	**19.58**	**24**	19.56	68	18.97
Instance 2	188	19.31	**9**	**19.35**	55	18.65
Instance 3	146	16.36	**8**	**17.42**	46	14.87
Instance 4	315	14.18	**8**	**16.78**	43	13.39
Instance 5	404	**19.03**	**8**	18.89	59	17.59

Au vu de ces premiers résultats, la méthode par programmation dynamique apparaît comme étant la plus rapide. La méthode par réduction itérative des fenêtres présente l'avantage d'être plus égalitariste puisqu'elle optimise en quelque sorte un critère de type *leximin*. Dans la suite, le but est d'utiliser ces méthodes de base au sein d'un algorithme plus général qui fait de la sélection et de l'ordonnancement d'observations pour une constellation de satellites.

Références

[1] P. H. Morris, R. A. Morris, L. Khatib, S. Ramakrishnan, and A. Bachmann. Strategies for global optimization of temporal preferences. In *CP*, pages 408–422, 2004.

[2] G. Peng, R. Dewil, C. Verbeeck, A. Gunawan, L. Xing, and P. Vansteenwegen. Agile earth observation satellite scheduling : An orienteering problem with time-dependent profits and travel times. *Computers & Operations Research*, 111 :84–98, 2019.

[3] V. F. Yu, P. Jewpanya, S.-W. Lin, and A.A.N. P. Redi. Team orienteering problem with time windows and time-dependent scores. *Computers & Industrial Engineering*, 127 :213–224, 2019.