Déploiement efficace de drones pour la collecte de données de capteurs mobiles

Christelle Caillouet, Frédéric Giroire, Tahiry Razafindralambo

Déploiement efficace de drones pour la collecte de données de capteurs mobiles

Christelle Caillouet[1] et Frédéric Giroire[1] † et Tahiry Razafindralambo[2]

[1] *Université Côte d'Azur, CNRS, Inria, I3S, France*
[2] *Université de la Réunion, LIM, France*

Dans cette article, nous présentons un modèle d'optimisation pour le problème de couverture de capteurs mobiles par des drones. Le but est de déployer des drones de manière à collecter efficacement des données depuis des capteurs mobiles vers une station de base au cours du temps. Nous proposons un modèle se résolvant par génération de colonnes. Les résultats montrent l'efficacité de la formulation par rapport à un programme linéaire classique et ouvrent des perspectives quant à une analyse plus fine de l'évolution des positions des drones au cours du temps.

Mots-clefs : programmation linéaire, génération de colonnes, réseaux de capteurs, drones

1 Le problème de collecte aérienne des données (ADCP)

Nous considérons l'application consistant en la surveillance de cibles mobiles (i.e. des capteurs), par une flotte de véhicules aériens autonomes (i.e. des drones) [EFPS14]. Les drones constituent un réseau backbone couvrant les cibles au sol et collectant l'information jusqu'à une station de base à travers des liens radio multi-sauts. Nous avons développé différents programmes linéaires afin de résoudre le problème de collecte aérienne des données (ADCP). Une première formulation mixte résout ADCP à l'optimal, mais ne passe pas à l'échelle de la taille des instances. Nous avons donc développé une formulation différente pour ADCP, qui dépend d'ensembles de positions choisies pour les drones, basée sur une méthode de décomposition et résolue par génération de colonnes. Deux problèmes auxiliaires (ou pricing) sont présentés afin d'accélérer la résolution du modèle. Nous étendons les travaux existants dans la littérature sur la couverture optimale de capteurs au sol par un ensemble de drones. Dans [CR17], les capteurs sont statiques alors qu'ici ils sont mobiles sur la zone à couvrir. Du point de vue de l'optimisation, la collecte des données d'un ensemble de points vers une entité centrale peut se modéliser principalement de deux manières. Premièrement, le problème peut être vu comme une tournée de véhicules où la station de base constitue le point de collecte, et chaque drone doit déterminer sa trajectoire afin de visiter toutes les cibles au sol une seule fois [BFP$^+$11]. Deuxièmement, ADCP peut se modéliser comme un problème d'ensemble couvrant correspondant à la sélection d'un sous-ensemble de nœuds reliés à tous les nœuds d'un graphe [PRS16]. Nous choisissons ici le deuxième modèle qui permet de prendre en compte l'altitude des drones dans un modèle en trois dimensions. Plus spécifiquement, puisque les cibles sont mobiles, nous considérons la version dynamique du problème de couverture [CDM82], minimisant le coût de déploiement et le délai de collecte. ‡

Ce travail présente une nouvelle formulation optimale du problème de couverture de cibles mobiles, que ce soit du point de vue appliqué ou théorique, optimisant la qualité de couverture et la connectivité entre les drones dans un environnement 3D. Combiner ces contraintes et objectifs est très important puisque cela permettra de réduire les coûts de déploiement des drones et accroître leur utilisation.

2 Formulation linéaire

Nous considérons un modèle discret dans lequel les positions des cibles sont estimées à différents intervalles de temps. Soit P l'ensemble des positions 3D possibles pour les drones. Une cible est dite couverte

†Travail partiellement soutenu par le programme ANR "Investments for the Future" sous la référence ANR-11-LABX-0031-01.
‡. Une version étendue de ce papier a été présentée à WiSARn 2018 [CGR18]

par un drone si la distance entre la projection de la position du drone au sol et la cible est inférieure au rayon d'observation du drone. Pour une collecte efficace, les drones sont connectés entre eux et avec une station de base b située au sol. Etant donné l'ensemble des cibles N^t, le rayon de couverture, et le rayon de communication des drones, nous introduisons une représentation de notre topologie par un graphe dynamique. A chaque intervalle de temps $t \in [0,T]$ est associé le graphe $G^t = (V^t, E^t)$ défini de la manière suivante :

Definition 1 *Au temps $t \in [0,T]$ de la période d'observation, la topologie est modélisée par le graphe orienté $G^t = (V^t, E^t)$ où :*
- $V^t = \{b\} \cup P \cup N^t$, *et*
- $E^t = \{(b,p),\ p \in P \text{ and } (x_p, y_p) = \min_{q \in P} d(b,q)\} \cup \{(p,q),\ p,q \in P \text{ and } R_p \geq D_{pq}\}$
  $\cup \{(p,n),\ p \in P, n \in N^t \text{ and } r_p^h \geq d(p,n)\}$.

Dans G^t, nous calculons un sous-ensemble de nœuds S correspondant aux positions choisies pour les drones. Cet ensemble doit être connecté et couvrir toutes les cibles de N^t. Pour chaque ensemble, nous définissons un coût associé C_S égal à la somme des distances 3D entre chaque position et la station de base.

Afin de résoudre efficacement ADCP, nous dérivons une nouvelle formulation du problème, appelée ADCP_CG, définie sur les ensembles de positions choisies pour les drones. Cette formulation nous permet de séparer le problème en deux parties : (i) un programme maître qui gère la mobilité et assure la couverture des cibles au cours du temps, et (ii) un programme auxiliaire qui génère des nouveaux ensembles de positions connectées, qui couvre les cibles à un temps donné. Soit $\mathcal{S}$ l'ensemble de tous les ensembles de positions possibles, de taille exponentielle. Le but est déviter l'énumération de tous les ensembles connectés de positions en utilisant la génération de colonnes.

2.1 Programme maître

Soit z_S^t (resp. χ_n^t) la variable binaire indiquant si l'ensemble S est sélectionné au temps t (resp. si la cible n est couverte au temps t).

$$\min \left(\sum_{t=0}^{T} \sum_{S \in \mathcal{S}} (z_S^t \cdot C_S) + \max_{t \in [1,T]} | \sum_{S \in \mathcal{S}} (z_S^t \cdot C_S) - \sum_{S' \in \mathcal{S}} (z_{S'}^{t-1} \cdot C_{S'}) | \right) \tag{1}$$

$$\sum_{S \in \mathcal{S}} z_S^t = 1, \forall t \in [0,T] \tag{2}$$

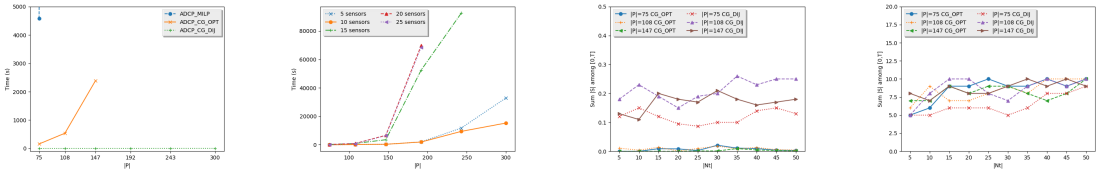
$$\chi_n^t \geq 1, \forall t \in [0,T], n \in N^t \tag{3}$$

$$\chi_n^t \leq \sum_{S \in \mathcal{S}} (z_S^t \cdot \sum_{u \in S} \lfloor \frac{r_u^h}{d_{un}} \rfloor), \forall t \in [0,T], n \in N^t \tag{4}$$

La fonction objectif (1) minimise le coût total de déploiement et la différence de coût entre deux temps consécutifs. Pour une meilleure coordination des drones, nous cherchons à les déployer le plus près possible de la station de base. Ils ont une plus courte distance à parcourir, ce qui limite l'énergie dépensée et le délai de collecte. Les contraintes (2) sélectionnent un seul ensemble connecté de positions de drones à chaque temps. Les contraintes (3) et (4) assurent que toutes les cibles sont couvertes à chaque temps. Pour cela, elles doivent se trouver dans la zone de couverture d'au moins un drone de l'ensemble S sélectionné.

2.2 Programmes auxiliaires

A partir du programme maître (1)-(4), des contraintes duales associées aux variables z_S^t, et des coûts réduits $\beta^{(i)}$ des contraintes (i), le programme auxiliaire détermine l'ensemble connecté de coût minimum (MWCS), où le coût correspond à $C_S \cdot (1-\gamma) - \sum_{n \in N^t} \beta_{nt}^{(4)} \sum_{p \in S} |\{(p,n)\} \cap E^t|$, avec γ correspondant à une fonction sur les variables duales $\beta^{(i)}$ qui dépendent de t. La génération de MWCS fournit soit un nouvel ensemble à ajouter aux variables du programme maître, soit certifie qu'aucune colonne existe. Si le coût de l'ensemble généré est inférieur à $\beta^{(2)}$, il est ajouté au programme maître. Nous avons développé une formulation mixte exacte (MILP) et une heuristique pour résoudre le programme auxiliaire.



(a) Temps de résolution. (b) Temps de résolution de ADCP_CG_OPT. (c) Précision des solutions de ADCP_CG. (d) Nb d'ensembles générés.

FIGURE 1: Temps de résolution et validation des solutions.

Formulation optimale : Une formulation optimale de MCWS calcule un flot entre la station de base et les cibles $n \in N^t$ dans le graphe G^t, assurant à la fois la couverture et la connectivité de l'ensemble généré à un temps donné. Mais ce programme est *NP*-difficile car il correspond au calcul d'un arbre de Steiner dont les nœuds qui le composent ne sont pas connus a priori.

Heuristique Dijkstra : Afin d'accélérer la résolution du programme auxiliaire, nous développons une heuristique basée sur l'algorithme de Dijkstra afin de générer un nouvel ensemble connecté de positions. Nous calculons itérativement les plus courts chemins entre la station de base et les cibles dans le graphe G^t au temps t, avec les coûts réduits sur les liens. Le chemin de plus fort coût ayant le premier nœud intermédiaire le plus haut est gardé à chaque temps et les positions des nœuds de ce chemin est enregistré dans l'ensemble résultat. Les coûts des liens de ce chemin sont fixés à 0 afin d'encourager les chemins suivants à utiliser les mêmes nœuds intermédiaires. Nous obtenons alors un ensemble vérifiant les propriétés voulues et vérifions s'il viole la contrainte duale afin de l'ajouter au programme maître.

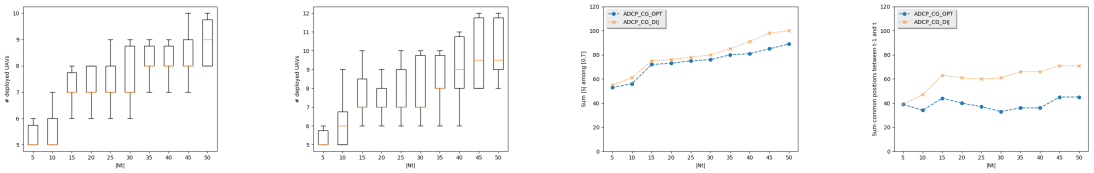
3 Resultats

Le modèle a été implémenté en Java et résolu en utilisant le solveur IBM Cplex 12.7.1. Les instances sont déployées sur une zone de $100m \times 100m$. Nous considérons 10 intervalles de temps pour la mobilité. Les coordonnées initiales, finales, et la vitesse de déplacement des cibles sont choisies aléatoirement dans la zone. A chaque étape, la position des cibles évolue en fonction de leur position précédente et de leur vitesse jusqu'à la destination. La zone de couverture est divisée en carrés égaux dans lesquels se trouve une position possible pour les drones en son centre. De cette manière, les sites candidats pour les drones forment une grille régulière. Pour chaque point de la grille, les altitudes possibles sont $\{10m, 25m, 45m\}$. La station de base est placée aux coordonnées $(0,0,0)$. Nous avons généré des instances comprenant 5 à 50 cibles, et entre 75 et 300 positions 3D pour les drones. Chaque drone a un angle de visibilité de 60 degrés et un rayon de communication de $30m$.

3.1 Performances de ADCP_CG

Dans la figure 1, nous validons notre modèle avec un programme auxiliaire optimal (ADCP_CG_OPT), et heuristique (ADCP_CG_DIJ). Notre modèle se résout pour les topologies testées avec plusieurs centaines de positions 3D, et donne des solutions entières très proches de l'optimal avec les colonnes générées. La formulation MILP classique (ADCP_MILP) prend déjà plusieurs milliers de secondes pour résoudre les petites topologies (Figure 1a), et ne donne plus de solution pour des topologies plus grandes dû à un problème de mémoire de Cplex. Au contraire, ADCP_CG résout les plus grandes instances jusqu'à 300 positions. Néanmoins, le temps de résolution de ADCP_CG_OPT augmente avec $|N^t|$ et l'augmentation est même exponentielle avec le nombre de positions (à cause de la complexité du programme auxiliaire optimal).

La figure 1c compare la qualité des solutions obtenue en comparant la valeur du programme maître entier résolu avec l'ensemble des colonnes générées, et la solution optimale de la relaxation linéaire du programme maître. Lorsque la courbe est à 0, la solution trouvée par notre modèle est optimale pour ADCP. Cet optimum est atteint pour un petit nombre de cibles (5 à 20), et pour différents nombres de positions. Lorsque l'optimum n'est pas atteint, nous voyons que la qualité des solutions reste très proche de l'optimal.



(a) Evolution du coût de déploiement (ADCP_CG_OPT et $|P| = 75$).

(b) Evolution du coût de déploiement (ADCP_CG_DIJ et $|P| = 75$).

(c) Somme de $|S|$ au cours du temps pour $|P| = 75$.

(d) Somme des positions communes entre 2 temps consécutifs ($|P| = 75$).

FIGURE 2: Validation de l'heuristique Dijkstra.

Enfin, le nombre d'ensemble générés durant le processus de génération de colonnes (Figure 1d) est très bas et toujours inférieur à 10 avec les différents pricing. Cela valide notre approche car nous n'avons besoin que d'un faible nombre d'ensembles (comparé à la taille exponentielle de $\mathcal{S}$) pour atteindre la solution optimale du programme maître relâché, nous donnant une borne inférieure sur l'optimum entier de ADCP.

3.2 Deploiement au cours du temps et validation de l'heuristique

Les figures 2a et 2b présentent le nombre de drones déployés au cours du temps. La valeur moyenne est représentée par une barre dans chaque boxplot. Pour chaque topologie, la variation du nombre de drones reste bornée par 3. ADCP_CG_DIJ génère des ensembles plus grands que ADCP_CG_OPT, mais le nombre additionnel de drones nécessaires est borné par 3. La figure 1a valide ADCP_CG_DIJ en terme de temps de résolution. Nous obtenons des solutions en moins de 2 secondes, même pour des topologies avec 50 cibles et 300 positions. De plus, la figure 1c présente la précision des solutions de ADCP_CG_DIJ et montre qu'elles se situent entre 10 et 20% de l'optimum. Elles utilisent plus de drones que ADCP_CG_OPT (Figure 2c), et les drones sont positionnés à une altitude plus importante. Néanmoins, les positions choisies par ADCP_CG_DIJ restent valides plus longtemps au cours du temps (Figure 2d). Ceci montre un avantage permettant de maintenir la collecte des données et maximiser la durée de vie du système aérien. En conclusion, ADCP_CG_DIJ obtient des solutions très précises, très rapidement.

4 Conclusion

Dans cet article, nous étudions le problème de la collecte aérienne de données de capteurs mobiles, à l'aide d'une flotte de drones. Nous présentons un modèle se résolvant par génération de colonnes, utilisant deux types de programmes auxiliaires. Notre modèle résout efficacement toutes les topologies testées. Le programme auxiliaire optimal étant *NP*-difficile, nous développons une heuristique permettant de résoudre de grandes instances en moins de 2 secondes avec un ensemble restreint de colonnes. Etant donné la littérature, c'est le premier modèle qui passe à l'échelle avec le nombre de capteurs et de positions 3D.

Références

[BFP+11] F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. L. Smith. Dynamic vehicle routing for robotic systems. *IEEE proceedings*, 99(9) :1482–1504, 2011.

[CDM82] J. W. Chrissis, R. P. Davis, and D. M. Miller. The dynamic set covering problem. *Applied Mathematical Modelling*, 6(1) :2 – 6, 1982.

[CGR18] C. Caillouet, F. Giroire, and T. Razafindralambo. Optimization of Mobile Sensor Coverage with UAVs. In *IEEE Infocom workshop WiSARN*, April 2018.

[CR17] C. Caillouet and T. Razafindralambo. Efficient Deployment of Connected Unmanned Aerial Vehicles for Optimal Target Coverage. In *IEEE GIIS*, 2017.

[EFPS14] J.J. Enright, E. Frazzoli, M. Pavone, and K. Savla. UAV routing and coordination in stochastic and dynamic environments. In *Handbook of Unmanned Aerial Vehicles*. Springer, 2014.

[PRS16] M. I. Popescu, H. Rivano, and O. Simonin. Multi-robot patrolling in wireless sensor networks using bounded cycle coverage. In *ICTAI*, Nov 2016.