



HAL
open science

Algorithmes de placement optimisé de drones pour la conception de réseaux de communication

Zacharie Alès, Sourour Elloumi, Adele Pass-Lanneau

► **To cite this version:**

Zacharie Alès, Sourour Elloumi, Adele Pass-Lanneau. Algorithmes de placement optimisé de drones pour la conception de réseaux de communication. Conference on Artificial Intelligence for Defense (CAID) 2022, DGA Maîtrise de l'Information, Nov 2022, Rennes, France. hal-03881005

HAL Id: hal-03881005

<https://hal.science/hal-03881005v1>

Submitted on 1 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Algorithmes de placement optimisé de drones pour la conception de réseaux de communication

Zacharie Alès
UMA
ENSTA Paris
Palaiseau, France
zacharie.ales@ensta-paris.fr

Sourour Elloumi
UMA
ENSTA Paris
Palaiseau, France
sourour.elloumi@ensta-paris.fr

Adèle Pass-Lanneau
CATOD
DGA

Abstract—L'établissement d'un réseau de télécommunications performant et robuste est un besoin opérationnel clé pour les forces armées en opérations. Dans les prochaines années, des drones haute altitude pourront être utilisés comme noeuds du réseau sur le théâtre. Une question est alors de décider de leur placement spatial. Dans ce travail, une approche de recherche opérationnelle est proposée afin d'obtenir des algorithmes efficaces de placement de drones.

Index Terms—drones haute altitude, conception de réseau, recherche opérationnelle, programmation linéaire en nombres entiers, heuristiques.

I. INTRODUCTION

Sur un théâtre d'opérations, les unités déployées au sol se déplacent et communiquent entre elles par des moyens de communication basés sur différentes technologies : UHF, VHF, 5G, etc. Dans un contexte de numérisation croissante du champ de bataille et afin d'augmenter les capacités de mise en réseau des unités sur le terrain, l'utilisation future de drones haute altitude (HAPS) est envisageable. Ces systèmes, positionnés dans la stratosphère, peuvent emporter des relais de communication de différents types. Ils sont ainsi un intermédiaire entre les moyens au sol et les moyens satellitaires.

Pour évaluer l'intérêt opérationnel de tels systèmes, il est nécessaire de savoir comment ils seraient déployés sur un théâtre, au profit des forces au sol. La question du déploiement porte à la fois sur la position géographique de ces vecteurs, et sur les relais de communication qu'ils devraient emporter.

L'optimisation du réseau formé par un ensemble de drones peut être représenté par un problème d'optimisation combinatoire, proche de problèmes classiques en recherche opérationnelle. Dans le cadre du projet AID "LoCHAPS" conjoint entre l'ENSTA Paris et la DGA, nous investiguons de nouvelles techniques de recherche opérationnelle, et notamment de programmation mathématique, afin de résoudre efficacement ce problème d'optimisation de réseau.

Les problématiques de placement de drones ont récemment attiré l'attention de la communauté académique, pour des applications duales et diversifiées (gestion de crise, déploiement de senseurs, ou déploiement d'un réseau de communication,

voir [1] pour une revue de littérature). Les outils développés relèvent souvent de l'algorithmique pour les systèmes embarqués, ou visent à obtenir un réseau auto-organisé de drones [2]. Dans les travaux présentés, nous nous posons à l'inverse la question du point de vue d'un optimisateur centralisé, et nous investiguons des outils algorithmiques de résolution exacte. Le but est de répondre à une question de dimensionnement à un niveau stratégique : de quels systèmes aura-t-on besoin, et en quelle quantité, pour répondre au besoin opérationnel ?

II. FORMALISATION DU PROBLÈME D'OPTIMISATION

Présentons plus en détail le problème considéré.

Éléments sur le théâtre. Nous considérons trois types d'éléments sur le théâtre : les unités, éventuellement une base et des drones déployés. Les unités sont des unités terrestres déployées sur le théâtre. On considère que l'on connaît leur position en fonction du temps. La base est une base avancée positionnée en un point du théâtre. Les drones sont déployés à une altitude fixée, en des points dont les coordonnées sont à optimiser. Ils ont une capacité d'emport déterminée par une charge utile et une puissance électrique maximale.

Relais de communication. On considère un ensemble de types de communications. Les unités et les drones sont munis de moyens de communications associés à ces différents types. Pour chaque unité, on dispose d'un rayon associé à chaque type de communication. Pour la base, on dispose similairement d'un rayon par type de communication. Pour ce qui est des drones, on dispose d'un ensemble de relais de communications, caractérisés par un rayon, un poids, et une puissance électrique. Un ensemble de relais forme un emport possible pour un drone si le poids total et la puissance totale n'excèdent pas la charge utile et la puissance maximale du drone, respectivement. Un drone muni de relais est ainsi doté de capacités de communication, représentées par les rayons de ses relais pour les différents types de communications.

Réseau recherché. Nous représentons la mise en réseau des différents éléments de la façon suivante. On considère des communications entre paires d'éléments drone-drone, drone-unité, drone-base. On dit qu'une paire d'éléments communique s'il existe au moins un type de communications pour lequel la distance entre les deux éléments est inférieure au rayon de communication des deux éléments. L'ensemble des éléments

qui communiquent entre eux forme ainsi un réseau connexe entre des drones, des unités et la base. Pour un réseau connectant uniquement un sous-ensemble d'unités, on dira que ces unités sont les unités *couvertes* par le réseau de drones.

Problème d'optimisation. Le problème d'optimisation considéré consiste à optimiser de façon jointe:

- les positions spatiales d'un nombre fixé p de drones ;
- les relais de communication qu'ils emportent ;

afin de former un réseau connexe entre la base, les p drones, et permettant de couvrir un sous-ensemble d'unités de taille maximale.

Ce problème comporte deux sous-problèmes distincts : le choix des relais emportés par les drones s'apparente à un problème de bin packing [3], tandis que le choix des positions des drones est un problème de couverture-connexité. Dans cet article, nous nous intéressons à ce second aspect.

III. PROBLÈME DE COUVERTURE-CONNEXITÉ

A. Définition

Nous étudions un cas particulier du problème précédent dans lequel on ne considère qu'un seul type de communication. Dans ce cas, chaque drone est doté d'un relais de communication et l'emport des drones n'est plus à optimiser. Seules les positions des drones sont à déterminer.

Plus précisément, le problème (Loc) que nous considérons est le suivant : *connaissant les coordonnées des unités et un ensemble de positions possibles pour une flotte de p drones identiques, connaissant aussi les rayons de couverture et de communication des drones, trouver les positions des drones qui permettent aux p drones de communiquer entre eux tout en couvrant un nombre maximum d'unités.*

Un exemple d'instance et une solution au problème (Loc) sont représentés dans la Figure 1. L'instance comporte 12 unités sur un terrain plat, représentées par les carrés. Les positions possibles pour les drones sont obtenues par discrétisation du plan de placement des drones : on considère comme positions possibles les points d'une grille de pas unitaire. Dans la solution, les $p = 4$ drones sont indiqués par les disques bleus. Le rayon de couverture des drones est égal à 1, le rayon de communication entre drones est égal à 2.5. Les liens de communications entre drones sont indiqués. La solution représentée permet de couvrir 8 unités sur 12.

Le problème (Loc) permet d'isoler la difficulté combinatoire principale du problème global, celle de la couverture-connexité. Des algorithmes développés pour (Loc) pourront ensuite être utilisés pour la résolution du problème global.

B. Littérature associée en Recherche Opérationnelle

Les problèmes de localisation et de couverture sont largement étudiés dans la littérature et ont de nombreuses variantes. Plusieurs problèmes classiques de Recherche Opérationnelle peuvent être vus comme des cas particuliers du problème (Loc).

Le problème de localisation simple [4], [5] consiste à ouvrir des dépôts d'une centrale d'achat parmi un ensemble d'emplacements possibles, puis de servir un ensemble de

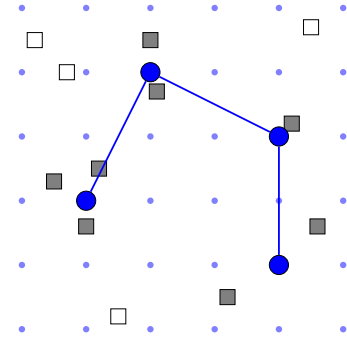


Fig. 1. Exemple d'instance et de solution au problème (Loc), représentées en vue de dessus. Les 4 drones sont représentés par les disques, les unités couvertes (resp. non couvertes) par les carrés gris (resp. blancs).

clients par le dépôt ouvert le plus proche de façon à minimiser la somme des coûts d'ouverture des dépôts et de service des clients. Une variante également très étudiée fixe le nombre de dépôts à ouvrir à un nombre k et s'appelle le problème des k -médians.

Concernant la connexité, un problème classique est celui de l'arbre de Steiner [6]. Il se pose dans un graphe non orienté avec des poids positifs sur les arêtes. On se donne aussi un sous-ensemble de sommets, généralement appelés terminaux. Le problème de l'arbre de Steiner consiste à sélectionner des arêtes qui relient tous les terminaux (en passant éventuellement par des sommets supplémentaires) et dont la somme des poids est minimale. Les arêtes d'une solution optimale forment donc un graphe sans cycle, soit un arbre.

Le problème de localisation connexe [7] associe ces deux derniers problèmes et a des similitudes avec le problème (Loc) puisqu'en plus d'ouvrir des dépôts, il faut qu'ils forment entre eux un graphe connexe, par exemple par des lignes de chemin de fer, de façon à ce qu'il soit toujours possible d'acheminer la marchandise d'un dépôt à un autre par le train.

Chacun de ces problèmes fait partie de la classe des problèmes NP-durs [3]. Ils trouvent des applications dans les télécommunications [7], dans la conception de réseaux de transport [8], ou dans le placement de capteurs [9]. Ils sont fréquemment traités dans la littérature par des techniques de Programmation Linéaire en Nombres Entiers, qui peuvent donner des résultats rapides même sur des instances très grandes, voir par exemple [5].

La problématique de placement de drones se distingue de la littérature par la nature a priori continue du placement des drones. En effet ceux-ci peuvent occuper n'importe quelle position dans un plan d'altitude fixée. On peut se ramener à un ensemble fini de positions par une discrétisation. Une telle discrétisation, si elle est fine, donne alors une instance du problème (Loc) avec un grand nombre de positions possibles.

IV. FORMULATIONS PLNE

A. Généralités

Afin de résoudre le problème (Loc), nous proposons deux formulations différentes représentées par des Programmes

Linéaires en Nombres Entiers (PLNE). Disposer de différentes formulations permet d'évaluer leurs performances numériques, et ainsi d'identifier la ou les formulations qui se comportent le mieux selon les instances à résoudre.

Une formulation PLNE peut être utilisée pour résoudre le problème par Branch & Bound, via des solveurs spécialisés (e.g., CPLEX). Cela nécessite cependant que la formulation soit de taille raisonnable, i.e., polynomiale en nombre de variables et contraintes. A l'inverse, les formulations de grande taille, par exemple avec un très grand nombre de contraintes, ne peuvent pas être directement traitées par les solveurs. Leur résolution se fait par un schéma algorithmique itératif, qui ajoute les contraintes parcimonieusement au fur et à mesure de la résolution grâce à une *séparation*. Ce schéma sera détaillé dans la suite, en Section IV-C.

Introduisons quelques notations pour le problème (Loc). On note \mathcal{U} l'ensemble d'unités qui sont indexées par u allant de 1 à $|\mathcal{U}|$ et \mathcal{P} l'ensemble de positions possibles des drones, indexées indifféremment par i ou j allant de 1 à $|\mathcal{P}|$. On connaît les coordonnées de chaque unité et de chaque position. On peut donc calculer la distance d entre n'importe quelle paire de points. Le nombre maximal de drones disponibles est p . Les drones sont tous identiques et ont un rayon de couverture R_{cov} (distance au-dessus de laquelle un drone ne peut pas communiquer avec une unité) et un rayon de communication R_{com} (distance au-dessus de laquelle un drone ne peut pas communiquer directement avec un autre drone).

Pour assurer la connexité entre les drones, on peut représenter la communication entre les drones par un graphe non orienté. Les sommets sont les positions où des drones peuvent être placés et une arête représente la possibilité de communication d'un drone vers un autre. On sait que le sous-graphe des positions occupées par des drones est connexe si et seulement s'il contient un arbre couvrant, soit une sélection connexe d'arêtes, avec autant d'arêtes que le nombre de sommets moins un.

Plusieurs définitions existent pour la notion d'arbre couvrant. L'une d'elles stipule que l'arbre couvrant, par une orientation de ses arêtes, forme une arborescence couvrante. Une telle orientation est équivalente à l'existence d'une numérotation des sommets où, pour tout arc (i, j) de l'arborescence, le numéro de i est strictement plus petit que le numéro de j . Cette numérotation est appelée un tri topologique. C'est cette définition de la notion d'arbre couvrant qui est utilisée dans la formulation MTZ, proposée Section IV-B. Une définition équivalente repose sur le fait qu'un arbre couvrant est exactement un sous-ensemble d'arêtes ne contenant pas de cycle et de taille égale au nombre de sommets couverts moins un. Cette définition est utilisée par la formulations sous-tours généralisés présentée Section IV-C.

B. Formulation MTZ

Ce type de formulation a été introduit dans [10] pour le problème du voyageur du commerce puis a été étendu à divers

problèmes de connectivité. Les variables de décision sont les suivantes :

- y_j variable binaire qui vaut 1 si et seulement si un drone est placé à la position $j \in \mathcal{P}$;
- x_{ij} , $i \neq j$ variable binaire qui vaut 1 si l'arc (i, j) est sélectionné pour faire partie de l'arborescence couvrante. Cela sous-entend que des drones sont placés en i et j et que la distance entre ces positions est inférieure au rayon de communication ;
- c_u variable binaire qui vaut 1 si et seulement si l'unité $u \in \mathcal{U}$ est couverte ;
- t_j numéro attribué à $j \in \mathcal{P}$ pour assurer l'existence d'un tri topologique.

La formulation (MTZ) pour le problème (Loc) s'écrit comme suit:

$$(MTZ) \left\{ \begin{array}{ll} \max & \sum_{u \in \mathcal{U}} c_u \\ \text{s.c.} & \\ & c_u \leq \sum_{j: d_{ju} \leq R_{cov}} y_j \quad u \in \mathcal{U} \quad (1) \\ & \sum_{j \in \mathcal{P}} y_j \leq p \quad (2) \\ & x_{ij} = 0 \quad i, j : d_{ij} > R_{com} \quad (3) \\ & x_{ij} \leq y_j \quad j, i \in \mathcal{P} \quad (4) \\ & \sum_{i \in \mathcal{P}} x_{ij} \leq y_j \quad j \in \mathcal{P} \quad (5) \\ & \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{P}} x_{ij} = \sum_{j \in \mathcal{P}} y_j - 1 \quad (6) \\ & t_j \geq t_i + 1 - p(1 - x_{ij}) \quad j, i \in \mathcal{P} \quad (7) \\ & y_j \in \{0, 1\} \quad c_u \in \{0, 1\} \\ & x_{ij} \in \{0, 1\} \quad t_j \geq 0 \end{array} \right.$$

L'objectif est de maximiser le nombre d'unités couvertes. Les contraintes (1) imposent que l'unité u soit identifiée comme non-couverte si aucun drone n'est placé aux positions qui peuvent la couvrir. La contrainte (2) assure qu'au plus p drones peuvent être déployés. Les contraintes (3) fixent à 0 les variables de communication x_{ij} entre des positions trop éloignées. Les contraintes (4) et (5) assurent qu'aucune communication n'est possible ni en sortie ni en entrée d'une position j non équipée de drone. Les contraintes (5) assurent de plus qu'au plus un arc entrant en j est sélectionné pour la position j , si elle est équipée d'un drone. La contrainte (6) impose que le nombre d'arcs dans l'arborescence couvrante soit égal au nombre de drones positionnés moins un. Enfin, les contraintes (7) assurent la cohérence du tri topologique. Si l'arc (i, j) est sélectionné alors $x_{ij} = 1$ et le numéro de j est au moins celui de i plus un. Sinon, on peut montrer que l'inégalité est toujours vérifiée puisqu'on n'a pas besoin d'attribuer plus de p numéros différents. La Figure 2 illustre les valeurs possibles de la variable t pour une configuration avec quatre drones positionnés.

C. Formulation sous-tours généralisés

Cette formulation est inspirée de la formulation de [11] pour le problème du voyageur de commerce. Dans cette formulation, on définit un arbre couvrant comme un choix d'arêtes ne

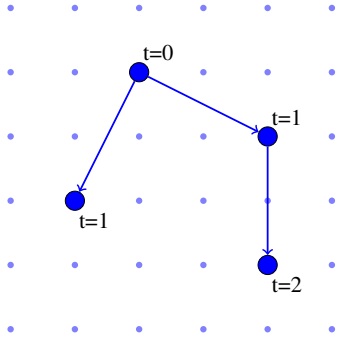


Fig. 2. Illustration de valeurs possibles des variables t , correspondant à des numéros selon un tri topologique.

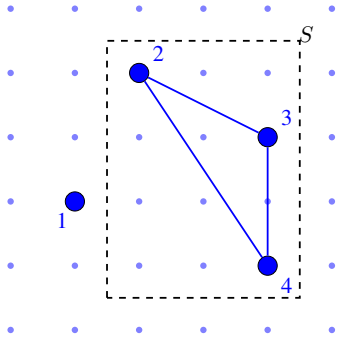


Fig. 3. Illustration de l'élimination des sous-tours. Les Contraintes (12) pour l'ensemble S ci-dessus et la position 2, imposent qu'au plus 2 arêtes soient sélectionnées à l'intérieur de S . Elles permettent ainsi d'éliminer le cycle constaté dans la solution ci-dessus.

comportant pas de cycle. Nous reprenons les mêmes variables y et c que ci-dessus. Pour les variables représentant les liens, on n'a pas besoin ici de les orienter. On considère donc une nouvelle variable binaire z_{ij} , $i < j$ valant 1 si et seulement si l'arête (i, j) est sélectionnée dans l'arbre couvrant.

La formulation (ST) pour le problème (Loc) est la suivante:

$$(ST) \left\{ \begin{array}{l} \max \sum_{u \in \mathcal{U}} c_u \\ \text{s.c.} \\ (1) - (2) \\ z_{ij} = 0 \quad i < j \in \mathcal{P} : d_{ij} > Rcom \quad (8) \\ z_{ij} \leq y_i \quad i < j \in \mathcal{P} \quad (9) \\ z_{ij} \leq y_j \quad i < j \in \mathcal{P} \quad (10) \\ \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{P} : i < j} z_{ij} = \sum_{j \in \mathcal{P}} y_j - 1 \quad (11) \\ \sum_{i \in S} \sum_{j \in S : i < j} z_{ij} \leq \sum_{j \in S} y_j - y_{j_0} \quad \forall S \subset \mathcal{P} : j_0 \in S \quad (12) \\ y_j \in \{0, 1\} \quad c_u \in \{0, 1\} \\ z_{ij} \in \{0, 1\} \end{array} \right.$$

Les contraintes (8)-(11) ont le même effet que les contraintes (3)-(6) de la formulation (MTZ). Les contraintes (12) interdisent à un ensemble de positions S de contenir un cycle, comme illustré dans la Figure 3.

Séparation des contraintes de sous-tours. La formulation (ST) utilise moins de variables que la formulation (MTZ) mais a un nombre exponentiel de contraintes (12). Pour

contourner cette difficulté et résoudre néanmoins cette formulation, on les ajoute au fur et à mesure que l'on constate leur violation par une solution $(\tilde{y}, \tilde{z}, \tilde{c})$. Une telle procédure est appelée *séparation*. Le schéma algorithmique pour résoudre (ST) est alors le suivant :

- résoudre la formulation (ST) avec seulement un sous-ensemble \mathcal{C} de contraintes (12), en déduire une solution $(\tilde{y}, \tilde{z}, \tilde{c})$;
- appliquer la séparation :
 - si aucune contrainte (12) n'est violée, alors $(\tilde{y}, \tilde{z}, \tilde{c})$ est une solution optimale de (ST) ;
 - sinon, ajouter la contrainte (12) violée à \mathcal{C} ;
- itérer.

Pour la séparation, il faut trouver un ensemble S et une position $j_0 \in S$ qui maximisent la violation de la contrainte associée (12). Plus formellement, ce problème peut être modélisé par le problème (SEP) suivant :

$$(SEP) \left\{ \begin{array}{l} \max \Delta = \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{P} : i < j} \tilde{z}_{ij} a_i a_j - \sum_{j \in \mathcal{P}} \tilde{y}_j a_j + \sum_{j \in \mathcal{P}} \tilde{y}_j h_j \\ \text{s.c.} \\ h_j \leq a_j \quad j \in \mathcal{P} \quad (13) \\ \sum_{j \in \mathcal{P}} h_j = 1 \quad (14) \\ a_j, h_j \in \{0, 1\} \end{array} \right.$$

où la variable binaire a_i vaut 1 si et seulement si la position i appartient à l'ensemble recherché S et la variable binaire h_j vaut 1 si et seulement si $j \in S$ et joue le rôle du j_0 recherché. Les contraintes (13) forcent h_j à valloir 0 si j n'est pas dans S . Les contraintes (14) assurent qu'une seule position j_0 sera sélectionnée. La fonction objectif mesure la violation maximale du point $(\tilde{y}, \tilde{z}, \tilde{c})$ par les contraintes (12). Ainsi, si on résout (SEP) et que l'on trouve, dans une solution optimale, $\Delta \leq 0$, cela signifie qu'aucune contrainte (12) n'est violée par ce point et le problème (ST) est résolu. Sinon, la solution (a, h) nous permet d'identifier un couple (S, j_0) correspondant à une contrainte (12) violée par le point. On ajoute cette contrainte et on continue le processus de résolution. Cet ajout dynamique d'inégalités violées peut être effectué à chaque nœud de la méthode and Branch & Bound et peut être implémenté avec un solveur comme CPLEX à l'aide des Callbacks.

V. RÉOLUTION HEURISTIQUE

Lorsque la taille du problème augmente, la résolution exacte des modèles PLNE présentés en section précédente peut nécessiter un temps de calcul prohibitif. Afin d'obtenir des solutions en temps limité pour ces instances de grande taille, nous introduisons dans cette section deux algorithmes gloutons. Nous en présentons, tout d'abord, une version déterministe puis, dans un second temps, une version randomisée. Nous verrons en Section VI que ces dernières permettent d'obtenir des solutions fournissant de bonnes performances en des temps de calcul très raisonnables.

A. Algorithmes déterministes

Le fonctionnement des algorithmes déterministes est illustré par l’Algorithme 1. Soit \mathcal{P} l’ensemble des positions possibles. On considère un sous-ensemble S de positions où sont placés des drones, initialement vide. A chaque itération, l’ensemble $\mathcal{P}_S \subseteq \mathcal{P} \setminus S$ des positions à portée d’au moins un des drones de S est identifié. Un drone est ensuite déployé dans une des positions de \mathcal{P}_S maximisant une fonction d’évaluation f . L’algorithme s’arrête lorsque S contient p positions, autrement dit p drones sont placés à ces positions, ou lorsqu’il n’existe plus aucune position à portée des drones de S .

Entrées :

- I : une instance du problème
- $f : \mathcal{P} \rightarrow \mathbb{R}^+$: fonction d’évaluation des unités

Sorties : Un ensemble S de positions formant un réseau connexe

```

1  $S \leftarrow \emptyset$ 
2  $\mathcal{P}_S \leftarrow \mathcal{P}$ 
3 tant que  $|S| < p$  et qu’il est possible d’ouvrir un site
   ( $\mathcal{P}_S \neq \emptyset$ ) faire
4    $s \leftarrow \operatorname{argmax}_{s \in \mathcal{P}_S} f(s)$ 
5    $S \leftarrow S \cup \{s\}$ 
6    $\mathcal{P}_S \leftarrow$  Sites de  $\mathcal{P} \setminus S$  à portée d’un site de  $S$ 
7 retourner  $S$ 

```

Algorithme 1 : Pseudo-code des heuristiques déterministes A_U et A_{PL} .

Nos algorithmes, nommés A_{PL} et A_U , se distinguent par leurs fonctions d’évaluation nommées f_{PL} et f_U , respectivement. L’algorithme A_{PL} se base sur la relaxation linéaire du programme (MTZ) défini dans la section précédente. La relaxation linéaire d’un PLNE correspond au problème obtenu en relâchant les contraintes d’intégrité (ex : $x \in \{0, 1\}$ deviendrait $x \in [0, 1]$). L’avantage de ce problème relâché est que sa résolution est beaucoup plus rapide, puisqu’il s’agit d’un Programme Linéaire (PL). Son inconvénient est qu’il fournit une solution généralement non réalisable car fractionnaire. L’information fournie par une solution fractionnaire peut néanmoins être exploitée et nous en tirons partie dans l’algorithme A_{PL} . A chaque itération d’ A_{PL} , on calcule la relaxation linéaire (c^*, t^*, x^*, y^*) de notre problème initial auquel ont été ajoutées des contraintes imposant l’ouverture des sites de S . La fonction d’évaluation est définie de la manière suivante :

$$f_{PL}(j) = y_j^*. \quad (15)$$

Ainsi, à chaque itération, A_{PL} ouvre un des sites $j \in \mathcal{P}_S$ pour lequel la valeur fractionnaire de la variable y_j^* est maximale.

Le second algorithme A_U ouvre à chaque itération un site permettant de maximiser le nombre de nouvelles unités couvertes :

$$f_U(j) = \begin{aligned} & \text{nombre d’unités couvertes par } S \cup \{j\} \\ & - \text{nombre d’unités couvertes par } S. \end{aligned} \quad (16)$$

Ces deux algorithmes sont dits déterministes puisque pour une instance donnée, ils fournissent toujours la même solution (sous réserve que la relaxation linéaire soit résolue de manière déterministe pour A_{PL}). Nous verrons que leurs temps de calcul sont très rapides en pratique mais que la qualité de leurs solutions n’est pas toujours satisfaisante.

B. Algorithmes randomisés

Afin de permettre d’améliorer les solutions au cours du temps, nous considérons également des versions randomisées A_{PL}^r et A_U^r de ces algorithmes représentées par l’Algorithme 2. Cet algorithme applique itérativement une version modifiée de l’Algorithme 1 (Instructions 4 à 11). La modification intervient à l’Instruction 7 qui au lieu de choisir le site maximisant f , effectue un tirage aléatoire pondéré par f . Cette étape permet d’apporter de la diversité dans les solutions en ne choisissant pas nécessairement à chaque étape le site le plus prometteur. Si la solution S obtenue est la meilleure trouvée jusqu’ici, S_{best} est mis à jour (Instructions 9 et 10). Ce processus est répété tant que le nombre maximal d’itérations n’est pas atteint.

Entrées :

- I : une instance du problème
- $f : \mathcal{P} \rightarrow \mathbb{R}^+$: fonction d’évaluation des sites
- i_{max} : nombre maximal d’itérations

Sorties : Un ensemble S_{best} de sites formant un réseau connexe

```

1  $S_{best} \leftarrow$  Solution de l’Algorithme 1
2  $i \leftarrow 1$ 
3 tant que  $i \leq i_{max}$  faire
4    $S \leftarrow \emptyset$ 
5    $\mathcal{P}_S \leftarrow \mathcal{P}$ 
6   tant que  $|S| < p$  et qu’il est possible d’ouvrir un
     site faire
7      $s \leftarrow$  tirage aléatoire dans  $\mathcal{P}_S$  pondéré par  $f$ 
8      $S \leftarrow S \cup \{s\}$ 
9      $\mathcal{P}_S \leftarrow$  Sites de  $\mathcal{P} \setminus S$  à portée d’un site de  $S$ 
10    si  $S$  couvre plus de clients que  $S_{best}$  alors
11       $S_{best} \leftarrow S$ 
12     $i \leftarrow i + 1$ 
13 retourner  $S_{best}$ 

```

Algorithme 2 : Pseudo-code des heuristiques randomisées A_U^r et A_{PL}^r .

VI. RÉSULTATS NUMÉRIQUES

A. Résolution exacte

Nous étudions les performances numériques des algorithmes proposés, et l’impact sur celles-ci du pas η de la grille utilisée pour discrétiser les positions des drones. Pour ces expériences nous utilisons un Intel(R) Core(TM) i7-7820HQ CPU @ 2.90GHz équipé de 15GByte de mémoire RAM.

Jeux de données. On considère un terrain carré de côté $C = 6000$. Les abscisses et ordonnées des positions et

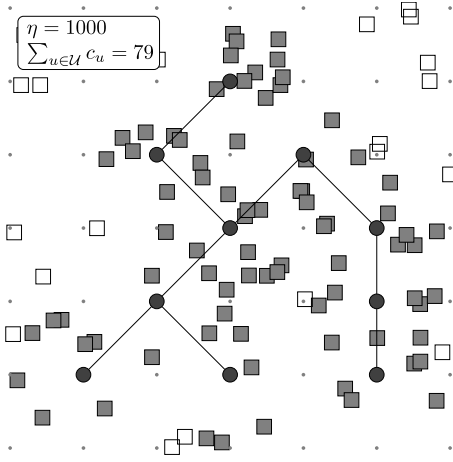


Fig. 4. Exemple de solution de l'instance $n = 100$, $p = 10$ et $\eta = 1000$, représentée en vue de dessus. Les drones sont représentés par les disques, les unités couvertes (resp. non couvertes) par les carrés gris (resp. blancs).

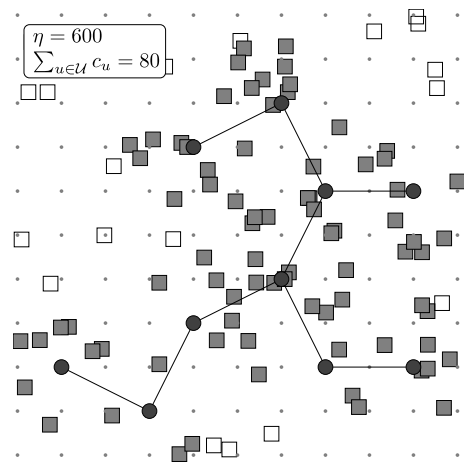


Fig. 5. Exemple de solution de l'instance $n = 100$, $p = 10$ et $\eta = 600$, représentée en vue de dessus. Les drones sont représentés par les disques, les unités couvertes (resp. non couvertes) par les carrés gris (resp. blancs).

des unités sont aléatoires dans l'intervalle $[0, C]$. Le nombre d'unités est $n = 100$ ou $n = 1000$. On dispose d'un modèle d'altimétrie du terrain, qui fournit l'altitude de chaque unité. On considère $p = 10$ drones, $R_{com} = R_{cov} = 1500$, placés à une altitude $Alt = 1200$. Le pas η de la grille de positions des drones est choisi comme un diviseur de C , au plus égal à R_{cov} . Nous avons choisi $\eta = 1500, 1200, 1000, 750, 600, 500, 400, 375, 300, 200$ ce qui donne un nombre de positions m respectivement égal à 25, 36, 49, 81, 121, 169, 256, 289, 441 et 961.

Solutions obtenues. La résolution de la formulation (MTZ) permet d'obtenir des solutions optimisées au problème (Loc), dont certaines sont illustrées dans les Figures 4,5,6. On observe dans ces trois figures qu'une réduction du pas η permet une augmentation du nombre d'unités couvertes $\sum_{u \in U} c_u$.

En pratique, il peut être souhaitable d'imposer la couverture d'un sous-ensemble d'unités prioritaires $\mathcal{U}_p \subset \mathcal{U}$ (e.g., les unités à l'avant du dispositif). Ceci peut aisément être intégré dans notre modélisation par l'ajout des contraintes :

$$c_u = 1 \quad \forall u \in \mathcal{U}_p \quad (17)$$

Dans les Figures 4,5,6 on peut par exemple observer que les cinq unités situées en haut à droite ne sont jamais couvertes. En imposant leur couverture, nous obtenons la solution représentée en Figure 7. La couverture des unités rouges se fait cependant au détriment du nombre total d'unités couvertes.

Performances de la formulation MTZ. Les résultats de la formulation MTZ sont donnés en Table I pour un nombre d'unité n égal à 100 et 1000. Afin d'accélérer le temps de résolution, la solution obtenue par A_U est donnée en entrée de la formulation.

Le temps de calcul est donné en secondes dans la colonne (Temps (s)). Il ne peut pas excéder un temps limite TL que nous avons fixé à 1800 secondes, soit 30 minutes. La colonne intitulée "gap" contient l'écart relatif entre la meilleure solution réalisable trouvée et la borne supérieure obtenue à l'issue

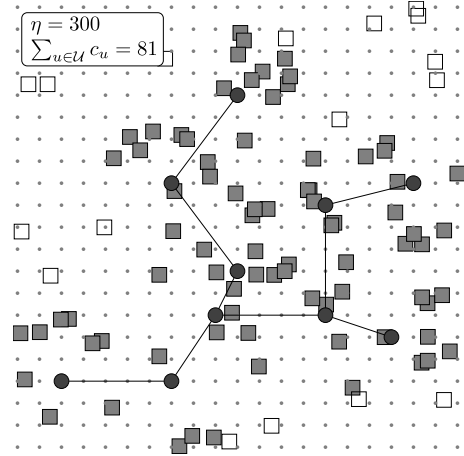


Fig. 6. Exemple de solution de l'instance $n = 100$, $p = 10$ et $\eta = 300$, représentée en vue de dessus. Les drones sont représentés par les disques, les unités couvertes (resp. non couvertes) par les carrés gris (resp. blancs).

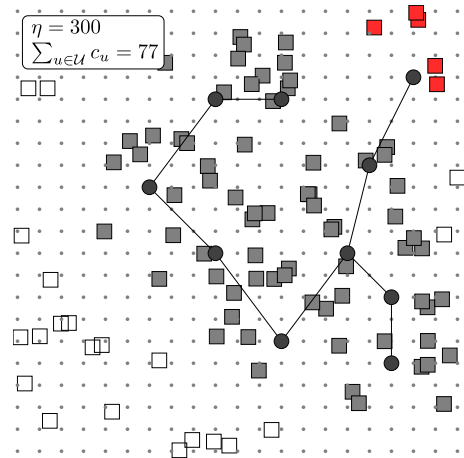


Fig. 7. Exemple de solution de l'instance $n = 100$, $p = 10$ et $\eta = 300$ en vue de dessus dans laquelle on impose que les unités représentées par les carrés rouges soient couvertes.

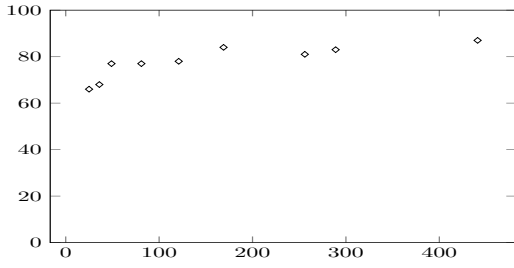


Fig. 8. Nombre optimal d'unités couvertes (%) selon le nombre de positions m .

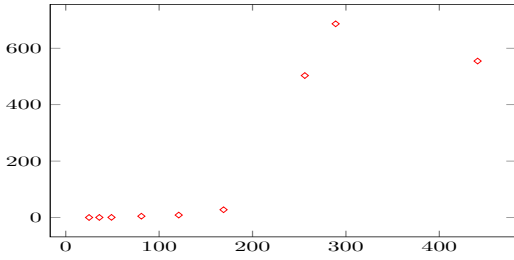


Fig. 9. Temps de calcul (secondes) selon le nombre de positions m .

de la résolution. Il est égal à 0 lorsque le problème est résolu optimalement en dessous du temps limite TL . Lorsque TL est atteint, ce gap permet de juger de la qualité de la solution trouvée à la fin de l'algorithme Branch & Bound. On observe que les instances ne sont pas résolues optimalement dès que l'on atteint 400 positions pour $n = 100$ et 600 positions pour $n = 1000$.

A priori, le positionnement réel des drones n'est pas contraint à des positions discrètes dans le plan d'altitude Alt . On recherche donc un pas η le plus petit possible, pour obtenir une discrétisation la plus fine possible. Afin d'estimer une taille de grille raisonnable, on peut s'intéresser à la valeur optimale du problème, c'est-à-dire le nombre maximal d'unités couvertes. Les Figures 8 et 9 montrent que le nombre maximal d'unités couvertes croît très lentement avec l'augmentation du nombre de positions au delà de 200 positions, tandis que le temps de calcul explose. Cela indique qu'une discrétisation comportant $m = 256$ positions correspondant à une grille 16×16 – qui pouvait a priori paraître grossière – peut en pratique offrir un bon compromis entre la valeur optimale et le temps de calcul.

Performances de la formulation (ST). Nous ne rapportons pas ici le détail des performances de la formulation (ST) car elles sont bien en-deçà de celles de la formulation (MTZ), comme on peut le voir sur l'exemple en Figure 10. Cette conclusion numérique est cohérente avec les résultats de la littérature [7]. Un travail algorithmique approfondi sur la séparation est nécessaire pour obtenir des temps de calcul probants pour (ST).

B. Résolution heuristique

La Table I permet également de comparer les résultats obtenus par la formulation (MTZ) et les heuristiques présentées en Section V. Le nombre d'itérations des algo-

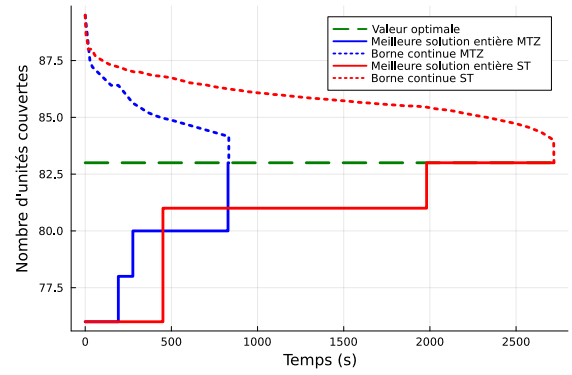


Fig. 10. Evolution des bornes inférieures et supérieures des formulations ST et MTZ au cours de la résolution pour l'instance $n = 100$, $p = 10$ et $\eta = 500$.

ritmes randomisés est fixé à 20. On constate tout d'abord que les heuristiques basées sur la relaxation continues A_{PL} et A_{PL}^r fournissent dans la plupart des cas des solutions de moins bonne qualité et des temps de calcul bien plus élevés que les heuristiques basées sur le nombre de clients couverts A_U et A_U^r . Ces temps de calcul peuvent s'expliquer par le fait que la taille de la formulation augmente rapidement lorsqu' η diminue rendant ainsi la résolution de la relaxation linéaire prohibitive. On peut également observer que les heuristiques randomisées permettent dans de nombreux cas d'augmenter le nombre d'unités couvertes. Ceci est d'autant plus intéressant pour A_U^r dont les temps de calcul restent très faibles.

Les heuristiques fournissent des solutions de bonne qualité mais rarement des solutions optimales. Il semble donc plus pertinent d'utiliser (MTZ) lorsque la taille du problème permet une résolution exacte. En revanche, pour les instances les plus grandes, A_U^r fournit les meilleurs résultats.

C. Compromis entre le nombre de drones et la capacité de couverture offerte

La résolution du problème (Loc) doit également permettre d'éclairer des choix sur le compromis à rechercher entre le coût du réseau de drones employé, et la capacité de couverture offerte par ce réseau.

On peut obtenir, pour différentes valeurs du nombre de drones p , le nombre maximal d'unités couvertes, et ainsi représenter le front de Pareto comme en Figure 11. On voit ainsi que la couverture de 80% des unités peut être assurée avec 9 drones, contre 15 drones pour couvrir 100% des unités.

D. Visualisation interactive de solutions

La résolution efficace du problème (Loc), soit par une méthode exacte soit par les heuristiques, fournit un placement de drones optimisé. Il est utile en pratique de visualiser les solutions obtenues pour les critiquer et éventuellement apporter des modifications à la modélisation du problème. Un exemple de visualisation de solution est donné Figure 12.

n	η	m	Nombre d'unités couvertes					Gap final (%)		Temps (s)				
			(MTZ)	A_{PL}	A_{PL}^r	A_U	A_U^r	(MTZ)	(MTZ)	A_{PL}	A_{PL}^r	A_U	A_U^r	
100	1500	25	72	72	72	72	72	0.0	0	0	0	0	0	
	1200	36	70	66	69	70	70	0.0	1	0	3	0	0	
	1000	49	79	24	71	71	73	0.0	1	0	4	0	0	
	750	81	79	18	70	72	74	0.0	6	1	18	0	0	
	600	121	80	56	76	77	77	0.0	100	1	27	0	0	
	500	169	83	14	62	76	81	0.0	772	3	96	0	1	
	400	256	85	14	66	83	83	2.7	TL	25	525	0	0	
	375	289	83	14	62	79	81	5.6	TL	19	774	0	1	
	300	441	82	13	60	81	81	10.2	TL	123	TL	0	1	
	200	961	79	45	45	79	80	17.4	TL	1072	TL	1	3	
1000	1500	25	607	607	607	604	607	0.0	0	0	2	0	2	
	1200	36	604	568	577	602	604	0.0	1	0	4	0	2	
	1000	49	671	361	635	665	665	0.0	2	0	11	0	1	
	750	81	692	292	537	690	692	0.0	17	1	28	0	2	
	600	121	634	347	428	634	647	10.2	TL	3	93	0	2	
	500	169	689	463	589	689	699	5.1	TL	5	269	0	3	
	400	256	665	267	489	665	685	11.5	TL	21	1042	0	3	
	375	289	683	424	434	683	690	9.0	TL	50	1327	0	2	
	300	441	676	386	437	676	721	12.9	TL	159	TL	0	4	
	200	961	686	248	249	686	703	13.8	TL	TL	TL	0	7	

TABLE I
COMPARAISON DES RÉSULTATS DES DIFFÉRENTES MÉTHODES DE RÉOLUTION. $TL = 1800$ SECONDES.

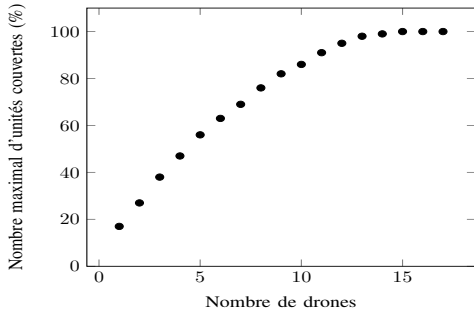


Fig. 11. Front de Pareto entre le nombre de drones déployés p et le nombre d'unités couvertes (%), pour $m = 256$ positions.

VII. EXTENSIONS

De nombreuses extensions du problème (Loc) peuvent être considérées afin de mieux représenter la réalité du terrain. L'intérêt des modélisations PLNE est d'être très souples à l'ajout de nouvelles contraintes ou critères. Parmi ces extensions, peuvent être prises en compte :

- les communications multi-technologies, les contraintes de charge utile et de puissance max des drones comme présentées Section II. Cette généralisation a été étudiée précédemment [12], et les algorithmes proposés pour (Loc) s'étendent naturellement.
- l'efficacité énergétique du réseau, par l'introduction d'un critère d'optimisation secondaire pénalisant les distances de communication drone-drone importantes.
- la géométrie du réseau, afin d'améliorer sa résilience. On peut par exemple imposer que les unités soient à portée de deux drones pour être couvertes, ou bien contraindre les variables x définissant les communications entre drones.

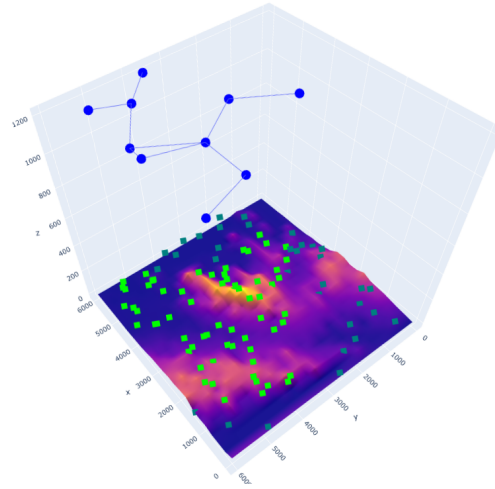


Fig. 12. Vue en 3D d'une solution obtenue, comprenant le modèle terrain, les unités couvertes en vert, et le réseau de drones en bleu.

VIII. CONCLUSION

Ces travaux ont permis d'identifier, au coeur de la problématique de placement de drones, un problème de couverture-connexité pour lequel les outils de Recherche Opérationnelle sont adaptés. Des approches algorithmiques exactes et heuristiques ont été proposées. Un point clé des performances des algorithmes est la discrétisation des positions possibles des drones. Les heuristiques proposées permettent d'obtenir des solutions très proches de l'optimum sur les instances considérées.

Une perspective est d'utiliser des techniques plus avancées de programmation mathématique pour résoudre, à l'exact, des instances de plus grande taille. L'extension des algorithmes présentés ici pour les cas plus généraux – multi-technologies, efficace en énergie, résilient – sera également poursuivie.

REFERENCES

- [1] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019.
- [2] K. Danilchenko, M. Segal, and Z. Nutov, "Covering users by a connected swarm efficiently," in *Algorithms for Sensor Systems*, C. M. Pinotti, A. Navarra, and A. Bagchi, Eds. Cham: Springer International Publishing, 2020, pp. 32–44.
- [3] B. H. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*. New York, NY: Springer-Verlag, 2012.
- [4] G. Laporte, S. Nickel, and F. S. da Gama, *Introduction to Location Science*. Cham: Springer International Publishing, 2015, pp. 1–18. [Online]. Available: https://doi.org/10.1007/978-3-319-13111-5_1
- [5] J.-F. Cordeau, F. Furini, and I. Ljubić, "Benders decomposition for very large scale partial set covering and maximal covering location problems," *European Journal of Operational Research*, vol. 275, no. 3, pp. 882–896, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0377221718310737>
- [6] I. Ljubic, R. Weiskircher, U. Pferschy, G. Klau, P. Mutzel, and M. Fischetti, "An algorithmic framework for the exact solution of the prize-collecting steiner tree problem," *Math. Program.*, vol. 105, pp. 427–449, 02 2006.
- [7] S. Gollowitzer and I. Ljubić, "MIP models for connected facility location: A theoretical and computational study," *Computers & Operations Research*, vol. 38, no. 2, pp. 435–449, 2011. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0305054810001334>
- [8] V. Bucarey, B. Fortz, N. González-Blanco, M. Labbé, and J. A. Mesa, "Benders decomposition for network design covering problems," *Computers & Operations Research*, vol. 137, p. 105417, 2022.
- [9] A. Murray, K. Kim, J. Davis, R. Machiraju, and R. Parent, "Coverage optimization to support security monitoring," *Computers, Environment and Urban Systems*, vol. 31, pp. 133–147, 03 2007.
- [10] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," *Journal of the ACM (JACM)*, vol. 7, no. 4, pp. 326–329, 1960.
- [11] G. Laporte, "Generalized subtour elimination constraints and connectivity constraints," *Journal of the Operational Research Society*, vol. 37, no. 5, pp. 509–514, 1986.
- [12] Z. Alès, S. Elloumi, M. Y. Naghmouchi, A. Pass-Lanneau, and O. Thuillier, "Planification optimisée du déploiement d'un réseau de télécommunication multitechnologie par dispositifs aéroportés sur un théâtre d'opérations extérieures," in *23ème congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision*. Villeurbanne - Lyon, France: INSA Lyon, Feb. 2022. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-03595379>