



HAL
open science

VESPA, ou l'art de coordonner une flotte de drone sans leader

Yann Busnel, Christelle Caillouet, David Coudert

► **To cite this version:**

Yann Busnel, Christelle Caillouet, David Coudert. VESPA, ou l'art de coordonner une flotte de drone sans leader. ALGOTEL 2020 – 22èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, Sep 2020, Lyon, France. pp.1-4. hal-02796530

HAL Id: hal-02796530

<https://inria.hal.science/hal-02796530>

Submitted on 5 Jun 2020

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.

VESPA, ou l'art de coordonner une flotte de drone sans leader

Yann Busnel¹, Christelle Caillouet² et David Coudert²

¹IMT Atlantique, IRISA, Rennes, France

²Université Côte d'Azur, Inria, I3S, CNRS, France

Fournir un accès à des services réseau à tout moment et en tout lieu est aujourd'hui une question cruciale, en particulier dans les situations d'urgence en cas de catastrophe. Une réponse naturelle à un tel besoin est l'utilisation de drones volants autonomes pour aider à découvrir les survivants et fournir une connectivité réseau aux équipes de secours sur place. Nous proposons VESPA (*Vehicle Spreading using Self-organized Parallel Algorithm*), un algorithme décentralisé utilisant uniquement les informations à un saut de chaque drone, afin de découvrir des cibles dont la localisation est initialement inconnue et s'auto-organiser pour assurer la connectivité entre elles et la station de base, à l'aide d'un réseau aérien multi-sauts sans fil. Nous prouvons que la connectivité, la terminaison et la couverture sont préservées pendant toutes les étapes de notre algorithme, et nous évaluons les performances de l'algorithme par des simulations. La comparaison avec un travail existant illustre l'efficacité de VESPA à la fois en termes de cibles découvertes et de nombre de drones utilisés.

Mots-clefs : Drone, Auto-organisation, Déploiement décentralisé, Gestion de crise, Connectivité réseau

1 Introduction

Les catastrophes naturelles telles que les tremblements de terre, les tsunamis, les tempêtes tropicales, les incendies, *etc.* représentent des domaines d'application important pour les véhicules aériens sans pilote (UAV), ou drones [ENCA17]. Dans de telles situations, le temps est un facteur critique pour trouver des survivants, et il est important de fournir une connectivité par le biais d'une solution rapide et efficace aux équipes de secours. L'utilisation de drones est devenue une solution clé pour améliorer la capacité de réaction aux catastrophes, faire face aux conditions environnementales extrêmes sans aucune connectivité réseau et effectuer des travaux de sauvetage en temps utile. L'objectif est de construire un système de drones autonomes pour participer aux opérations de recherche et de sauvetage, fournir les services requis ou collecter des informations surveillées en permanence depuis le sol. Par exemple, les drones ont été conçus pour exploiter les dispositifs intelligents appartenant aux survivants dans les zones dévastées et pour fournir une infrastructure de communication à la demande pour les opérations de sauvetage [MM18]. L'auto-organisation grâce à une planification efficace et décentralisée de la flotte de drones est l'étude prioritaire pour fournir un système efficace de gestion des catastrophes [BCC19b].

Dans cet article, nous considérons le problème suivant : étant donné un ensemble de cibles dont on ne connaît pas l'emplacement, le but est d'explorer la zone avec des drones pour découvrir autant de cibles que possible tout en assurant la connectivité avec une station de base (ou *puits*) pour permettre les échanges d'informations entre le puits, les drones et les cibles. La connectivité du réseau pour les communications air-sol et air-air doit être assurée par le déploiement tout en maximisant la zone surveillée par les drones.

La couverture des points d'intérêt (PoI) a également été étudiée dans les réseaux de capteurs mobiles sans fil (M-WSN) [HSDH19]. Dans ces réseaux, l'objectif est de s'assurer que les capteurs mobiles déployés fournissent la couverture requise pour la zone d'intérêt, tout en garantissant la connectivité du réseau déployé. Dans le cadre d'une approche décentralisée, les capteurs sont déployés de manière à surveiller l'environnement le plus longtemps possible et à couvrir une surface aussi grande que possible. L'algorithme *SaS* (*Spread and Shrink*) [REZN17] correspond aux travaux les plus proches des nôtres et utilise uniquement les informations environnantes et les interactions locales avec les robots mobiles à portée. SaS met en œuvre à la fois la phase de découverte et la couverture.

2 Modèle

Nous considérons une population \mathcal{D} de drones dans l'espace tridimensionnel. Le drone d_i , $i \in \{1, \dots, |\mathcal{D}|\}$, est situé à la position $p_i = (x_i, y_i, z_i)$, où (x_i, y_i) est la position dans l'espace 2D et z_i son altitude. Le puits \mathcal{S} est situé à $p_S = (x_S, y_S, z_S)$. Chaque drone d_i est équipé d'une antenne omnidirectionnelle lui permettant de communiquer avec d'autres drones situés dans une boule de rayon R_{p_i} (e.g., de l'ordre de 20 mètres). Soit \mathcal{N}_i l'ensemble de tous les drones accessibles dans la boule centrée sur d_i .

Nous considérons la présence d'un ensemble de cibles sur la zone surveillée correspondant à des survivants à détecter sur le terrain. La position fixe $p_{\tau_i} = (x_{\tau_i}, y_{\tau_i}, 0)$ de la cible $\tau_i \in \mathcal{T}$ est inconnue des drones et du puits. Une cible est couverte par un drone si elle est dans la zone d'intersection de la boule de communication du drone avec le sol. Pour éviter les trous de détection et assurer une couverture optimale, nous utilisons une tessellation triangulaire. Chaque drone est situé au centre d'un hexagone, et la portée de communication R_{p_i} lui permet de communiquer avec les drones situés au centre des hexagones voisins dans cette tessellation (Figure 1). Nous supposons aussi que la zone de couverture du drone circonscrit l'hexagone pour assurer une couverture complète de la zone.

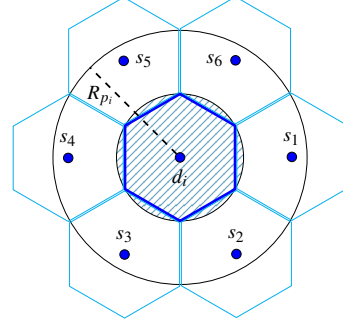


FIGURE 1: Tessellation triangulaire.

3 Algorithme VESPA

Tous les drones $d_i \in \mathcal{D}$ commencent à la position p_S correspondant à la station de base centralisant les opérations de recherche et de sauvetage. Les positions géographiques absolues sont inconnues (pas de GPS), mais les drones peuvent déduire leurs positions relatives entre eux. Chaque drone exécute indépendamment l'algorithme en suivant une alternance de 3 phases : (i) expansion, (ii) recouvrement, et (iii) équilibrage. Au cours du processus, chaque drone peut prendre 3 états possibles :

- 🔴 **inamovible** : ces drones resteront toujours dans leur position actuelle. Ils sont essentiels pour assurer la connectivité des cibles découvertes et la trajectoire aérienne vers le puits ;
- 🟢 **frontalier** : ces drones entourent la zone couverte, représentant le bord de la zone d'expansion ;
- 🔵 **libre** : ces drones peuvent se déplacer librement, i.e., ils ne sont ni inamovibles ni frontaliers.

Pour assurer la connectivité pendant les phases d'expansion successives, nous permettons aux drones d'être simultanément inamovibles et frontaliers, comme le montrent les sections suivantes. Pour des raisons d'espace, la description formelle de VESPA est disponible dans [BCC19a].

Phase d'expansion. Dans la phase d'expansion, les drones s'éloignent du puits pour détecter les cibles. Initialement, tous les drones sont à l'état **libre** à la position p_S . A chaque étape de cette phase, chaque drone libre situé au centre s_0 d'un hexagone décide indépendamment s'il se déplace vers l'un des centres s_1, \dots, s_6 des hexagones voisins (figure 1) ou s'il reste en s_0 . Si plusieurs drones se trouvent à s_0 , l'algorithme garantit qu'au moins un drone restera à cette position pour maintenir la couverture et la connectivité.

Nous proposons une procédure de terminaison, permettant de passer à la phase suivante de l'algorithme. En fin d'expansion, tout drone n'étant pas entouré de 6 autres drones, basculera en état **frontalier**.

Phase de recouvrement. À la fin de la phase d'expansion, chaque zone est soit vide, soit peuplée par un seul drone. Chaque drone situé sur un spot comprenant une cible, devient **inamovible** et cherchera à la fois à créer un chemin de communication multi-hop le plus court vers le puits, et à se connecter au drone frontalier le plus proche pour assurer la connexité de l'ensemble du réseau. Les drones situés sur ces chemins passent alors en état **inamovibles**, créant un arbre recouvrant connexe ayant le puits pour racine. Comme précédemment, une procédure de terminaison est proposée pour passer à la phase suivante.

Ainsi, à la fin de la phase de recouvrement, au moins un chemin de drones inamovibles existe entre le puits et la frontière de la zone explorée.

Phase d'équilibrage. Seuls les drones libres sont autorisés à se déplacer dans cette phase. Chaque drone libre choisit au hasard une zone voisine qui l'éloigne du puits. Ce processus est répété jusqu'à ce que tous les drones atteignent un endroit occupé par un drone frontalier. Nous équilibrons alors le nombre de drones

VESPA, ou l'art de coordonner une flotte de drones sans leader

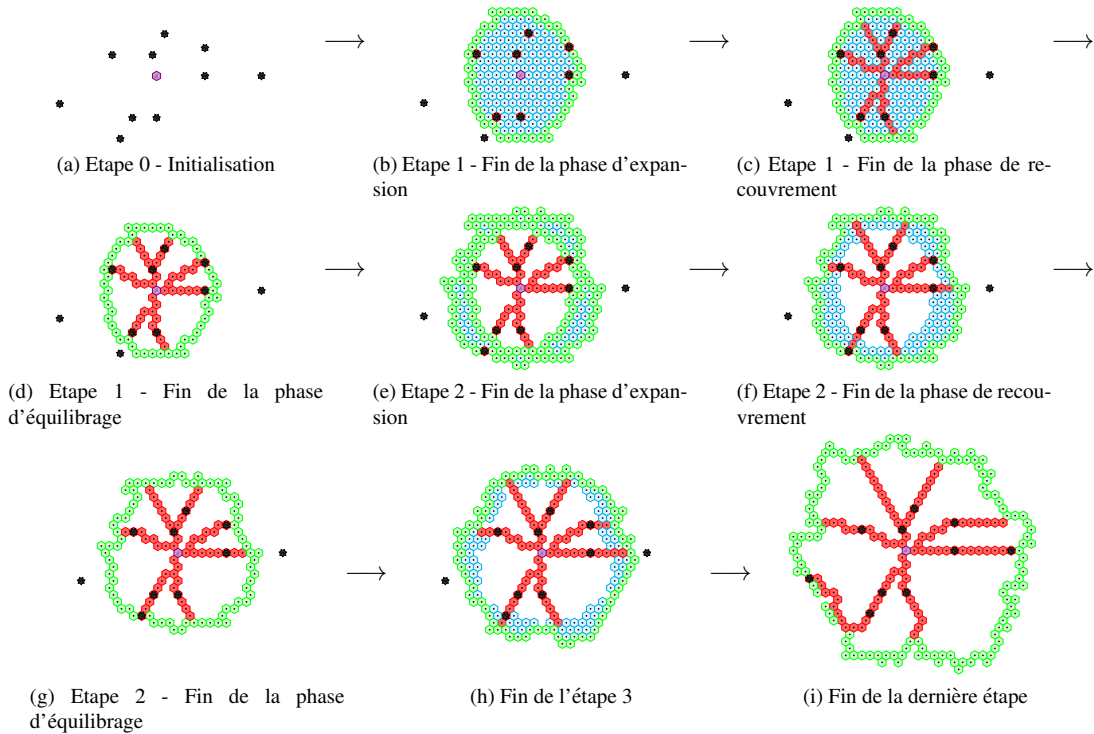


FIGURE 2: Exemple d'exécution de l'algorithme VESPA, avec 10 cibles et 217 drones.

par point de la frontière de sorte que le nombre de drones sur deux points voisins de la frontière diffère de un au maximum. Lorsqu'un drone libre atteint un drone frontalier, il suit le même processus que lors de la phase d'expansion, permettant de détecter la terminaison de cette phase.

Enfin, nous répétons les 3 phases expansion, recouvrement et équilibrage, jusqu'à ce que tous les drones soient *inamovibles* ou *frontaliers*. Le lecteur intéressé trouvera dans [BCC19a] l'ensemble des preuves formelles attestant du bon fonctionnement de VESPA. Nous y démontrons que chaque phase respecte les garanties de connectivité, de couverture et de terminaison.

Exemple. Nous illustrons dans la figure 2 le déroulement de VESPA, phase par phase, sur un exemple avec 217 drones et 10 cibles à couvrir (représentées par des étoiles noires). Initialement, tous les drones sont à l'état *libre* et se trouvent sur le puits (Figure 2a). Le processus commence par une première phase d'expansion, permettant de couvrir la zone la plus large possible centrée sur le puits. A la fin de la phase, tous les drones situés en bordure de la zone d'expansion changent d'état pour devenir *frontalier* (Figure 2b). Puis, les drones couvrant une zone comprenant une cible passent à l'état *inamovible* et lancent la phase de recouvrement, afin de garantir la connectivité entre le puits, les cibles détectées et la frontière (Figure 2c). La 3ème phase de VESPA permet ensuite à tous les drones encore en état *libre* d'atteindre la frontière, afin de préparer la prochaine phase d'expansion (Figure 2d). Les figures 2e à 2i illustrent la poursuite de l'exécution de VESPA, nécessitant 11 étapes supplémentaires pour atteindre l'expansion maximale. Une fois la dernière étape atteinte (plus aucun drone n'est en état *libre*), tous les drones non nécessaires à la connectivité des cibles avec le puits retournent à leur point de départ. Nous pouvons observer que la Figure 2c illustre également l'état du système à la terminaison de SaS, l'algorithme de référence de la littérature [REZN17].

4 Validation et évaluation des performances

Nous avons simulé le modèle distribué de notre algorithme dans le scénario suivant. Un puits est situé au centre de la zone à surveiller. Nous déployons au hasard autour du puits des cibles dont les coordonnées sont inconnues des drones et du puits. Nous exécutons l'algorithme sur des milliers de topologies aléatoires,

faisant varier le nombre de cibles de 1 à 30, et le nombre de drones de 10 à 250. Comme pour SaS [REZN17], nous fixons la portée de communication R_{p_i} de tous les drones à 20m. Nous validons VESPA par des simulations et comparons ses performances avec SaS en termes de nombre de cibles découvertes, et de nombre de drones nécessaires pour couvrir et assurer la connectivité avec les cibles.

Pour assurer une comparaison équitable avec le SaS, nous avons reproduit les mêmes scénarios que dans [REZN17]. Nous exécutons VESPA avec un nombre variable de drones et représentons le nombre de cibles découvertes dans la figure 3. La découverte de toutes les cibles dépend du nombre de drones. En effet, plus il y a de drones, plus la zone explorée est grande, et donc plus le nombre de cibles découvertes est important. Comme VESPA garantit la connectivité et la couverture à chaque étape du processus, le nombre de drones libres permettant de se déplacer et d'élargir la zone surveillée diminue avec le temps.

Cependant, même avec très peu de drones (20), il est possible de découvrir au moins une cible. La figure 3 montre également qu'il est possible de découvrir toutes les cibles avec 70 drones. Avec 160 drones, les cibles sont toujours découvertes, et seulement dans 99% des scénarii avec 120 drones. Ce comportement montre que le caractère aléatoire des décisions de mouvement est géré et contrôlé par VESPA. De plus, avec un même nombre de drones, VESPA découvre en moyenne deux à trois fois plus de cibles que SaS (par exemple, 3 fois plus avec 50 drones et 2,5 fois plus avec 100).

Nous comparons maintenant les résultats sur des topologies aléatoires où les cibles sont déployées au hasard sur la zone surveillée, d'environ 6,3 km². La figure 4 représente le nombre de drones nécessaires pour couvrir toutes les cibles. Le nombre de drones requis augmente avec le nombre de cibles pour les petites valeurs : 45 drones doivent se déployer sur la zone pour couvrir une cible, tandis que 90 drones sont nécessaires pour découvrir 10 cibles en moyenne. Ce phénomène se retrouve également avec SaS, alors que le nombre de drones nécessaires est plus important (deux fois plus de drones pour 5 cibles). Lorsque le nombre de cibles est important, le nombre de drones est toujours inférieur à 140. En effet, les chemins calculés par VESPA regroupent plusieurs cibles sur un même chemin, limitant ainsi le nombre de drones supplémentaires requis dans la solution. VESPA réduit de 40 % le nombre de drones nécessaires pour découvrir autant de cibles que SaS, tout en optimisant la taille des trajectoires se connectant au puits.

Références

- [BCC19a] Y. Busnel, C. Caillouet, and D. Coudert. Self-organized Disaster Management System by Distributed Deployment of Connected UAVs. In *ICT-DM*, Paris, France, 2019.
- [BCC19b] Y. Busnel, C. Caillouet, and D. Coudert. Self-organized UAV-based Supervision and Connectivity : Challenges and Opportunities. In *NCA*, Cambridge, MA, USA, 2019.
- [ENCA17] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz. Help from the sky : Leveraging uavs for disaster management. *IEEE Pervasive Computing*, 16(1) :24–32, Jan 2017.
- [HSDH19] Hailong Huang, Andrey V. Savkin, Ming Ding, and Chao Huang. Mobile robots in wireless sensor networks : A survey on tasks. *Computer Networks*, 148 :1 – 19, 2019.
- [MM18] F. Mezghani and N. Mitton. Open problem : Energy-and time-efficient dynamic drone path planning for post-disaster network servicing. In *ODS*, Taormina, Italy, 2018.
- [REZN17] T. Razafindralambo, M. Erdelj, D. Zorbas, and E. Natalizio. Spread and shrink : Point of interest discovery and coverage with mobile wireless sensors. *JPDC*, 102(C) :16–27, 2017.

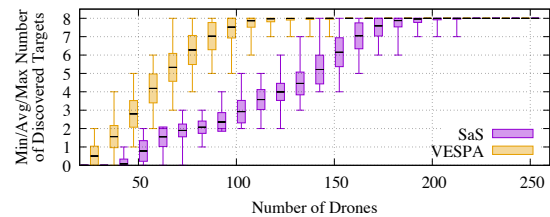


FIGURE 3: Boîte à moustaches du nombre de cibles découvertes, avec les valeurs min, max, moyenne et écart type.

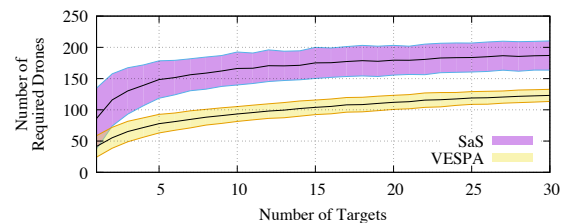


FIGURE 4: Nombre moyen de drones nécessaires pour découvrir toutes les cibles positionnées au hasard.