



HAL
open science

Résolution multicritère socialement acceptable du problème de réparation des contrats 4d dans le cadre de la gestion du trafic aérien sans pilote

Youssef Hamadi, Gauthier Picard

► To cite this version:

Youssef Hamadi, Gauthier Picard. Résolution multicritère socialement acceptable du problème de réparation des contrats 4d dans le cadre de la gestion du trafic aérien sans pilote. 10^{ème} Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle, AFIA-Association Française pour l'Intelligence Artificielle, Jul 2024, La Rochelle, France. hal-04642267

HAL Id: hal-04642267

<https://hal.science/hal-04642267>

Submitted on 26 Jul 2024

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.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Résolution multicritère socialement acceptable du problème de réparation des contrats 4D dans le cadre de la gestion du trafic aérien sans pilote

Youssef Hamadi¹, Gauthier Picard²

¹ Tempero, France

² DTIS, ONERA, Université de Toulouse, France

youssefh@tempero.tech
gauthier.picard@onera.fr

Résumé

Nous utilisons des systèmes multi-agents pour résoudre les conflits entre les trajectoires de vol planifiées (contrats 4D) des drones dans le trafic urbain. Cela permet de garantir la sécurité et la qualité du service tout en tenant compte de la facilité du contrôle humain. Nous explorons différents mécanismes de prise de décision (recherche dans les graphes, enchères, etc.) et critères d'évaluation (délai, qualité, énergie) pour ces agents. Nous simulons un scénario de surveillance avec des perturbations pour tester l'efficacité de ces approches.

Mots-clés

Gestion du trafic aérien sans pilote; drone; Coordination; DCOP; Enchères SSI; Optimisation multicritère

Abstract

We use multi-agent systems to solve conflicts between planned flight paths (4D-contracts) of drones in urban traffic. This ensures safety and service quality while considering ease of human control. We explore different decision-making mechanisms (graph search, auctions, etc.) and evaluation criteria (delay, quality, energy) for these agents. We simulate a surveillance scenario with disruptions to test the effectiveness of these approaches.

Keywords

Unmanned Air Traffic Management; UAV; Coordination; DCOP; SSI Auctions; Multi-Criteria Optimization

1 Introduction

La mobilité aérienne urbaine (UAM) vise à créer des systèmes de transport sûrs utilisant des aéronefs électriques innovants pour le transport de passagers ou de marchandises dans les zones urbaines et suburbaines. Sa réussite nécessite une stratégie cohérente, capable d'harmoniser efficacement l'efficacité et la sécurité. Ce nouveau système fait face à plusieurs défis, tels que le développement d'infrastructures, l'intégration avec les systèmes de gestion du trafic aérien existants, les obstacles réglementaires et l'acceptation publique. L'acceptation publique de la gestion du trafic aérien sans pilote (UTM) est la mesure dans laquelle les individus et les communautés sont prêts à accepter et à soutenir l'intégration des aéro-

nefs sans pilote dans leur vie quotidienne. Elle est influencée par des facteurs tels que la sécurité, la protection de la vie privée et l'impact environnemental. L'acceptation publique est cruciale pour l'adoption à grande échelle des systèmes UTM, d'où l'importance de développer des technologies qui la favorisent. Dans le domaine de l'UTM, la gestion du trafic aérien (ATM) est un défi majeur. Elle se caractérise par ses complexités techniques et son rôle essentiel dans la réponse à la préoccupation principale de l'acceptation publique, à savoir garantir la sécurité [25].

Contrairement aux systèmes traditionnels conçus pour les aéronefs conventionnels à voilure fixe opérant sur de longues distances à des altitudes élevées sous la surveillance de contrôleurs aériens, l'ATM doit superviser efficacement une multitude de petits aéronefs souvent hautement automatisés ou autonomes opérant à des altitudes plus basses dans un espace aérien complexe et encombré. Ces caractéristiques nécessitent un changement d'approche substantiel, soulignant la transition vers des systèmes ATM de plus en plus autonomes et pilotés par logiciel. Le souci de la sécurité, principale exigence d'acceptation par le public, impose de nouvelles contraintes à ces systèmes. En particulier, la clarté et la compréhensibilité des solutions de déconfliction deviennent essentielles et jouent un rôle crucial dans la confiance que le public accorde à ces systèmes avancés.

La première étape vers la définition de systèmes ATM intégrés pilotés par logiciel a commencé en 2018 avec l'Administration fédérale de l'aviation des États-Unis (FAA). Cela a marqué l'introduction du premier concept mondial d'opérations (ConOps) conçu pour la gestion du trafic des systèmes d'aéronefs sans pilote (UTM) [6]. L'UTM représente le cadre de gestion permettant de faciliter les opérations multiples de drones menées au-delà de la ligne de visée visuelle (BVLOS) dans des zones où les services traditionnels de trafic aérien ne sont pas disponibles.

Dans le cadre du système UTM, illustré dans la Figure 1, une relation de collaboration existe entre les opérateurs de drones et la FAA pour déterminer et transmettre les conditions de l'espace aérien en temps réel. La FAA propose des restrictions et des paramètres instantanés aux opérateurs d'UAS (Unmanned Aircraft Systems), qui assument la responsabilité de gérer leurs opérations en toute sécurité dans le cadre de ces lignes directrices, sans services directs de contrôle du trafic aérien de la part de la FAA. La communication et la coordination entre la FAA, les opérateurs de drones et les autres parties

objectif : elle sert d'outil pour modéliser et résoudre les conflits tactiques tout en offrant des informations précieuses pour évaluer les futures infrastructures potentielles pendant la phase de conception. Ils exploitent le concept des corridors UAM pour établir une définition mathématique précise de la séparation des véhicules [23]. Leurs solutions visent à minimiser l'écart global par rapport aux plans de vol programmés, nécessaire pour éviter les violations de la séparation par paire. Cette stratégie de déconfliction intègre à la fois des ajustements en vol, réalisés par des modifications de vitesse, et des retards au sol impliquant des décollages programmés. Pour évaluer leur approche, les auteurs mènent des expériences impliquant divers scénarios de conflit dans des topologies synthétiques de réseaux UAM.

Dans leur étude, [2] s'intéresse à l'avenir des services de taxis aériens urbains et introduit des méthodes mathématiques et des algorithmes novateurs visant à réaliser des opérations efficaces à grande échelle. Leur approche englobe deux aspects essentiels. Premièrement, ils adoptent une perspective centrée sur le passager qui introduit des classes de voyage et facilite l'échange d'informations entre les différents modes de transport, améliorant ainsi la qualité des services et garantissant des trajets multimodaux plus fluides. Deuxièmement, ils proposent une solution complète pour l'acheminement et la recharge des vols, qui non seulement minimise les coûts opérationnels immédiats, mais prolonge également la durée de vie de la batterie à long terme en réduisant les recharges gourmandes en énergie. En particulier, leur recherche se penche sur l'espace de conception des services de taxis aériens, en découvrant des solutions pour répondre aux problèmes d'équité dissimulés.

2.2 Approches décentralisées

L'étude [10] se concentre sur le modèle Sendai 2030, une projection réaliste de l'utilisation des drones pour les livraisons dans une zone spécifique du Japon. Ce cas implique jusqu'à 21 000 demandes d'opérations UAS sur une durée de service de 13 heures, ce qui représente un défi pour les méthodes de détection et de résolution des conflits avant le vol (CDR). Les auteurs proposent une méthode de réservation de l'espace aérien utilisant des trajectoires 4D et transposent le problème de la CDR avant le vol en un problème de recherche de chemin multi-agents (MAPF). Ils évaluent à la fois le traitement des demandes d'opérations UAS selon le principe du premier arrivé, premier servi (FCFS) et le traitement "par lots" et comparent leur débit. [14] s'attaquent au défi de la planification des trajectoires dans des scénarios où les demandes sont soumises par des UAS gérés par différents opérateurs. Ils introduisent un nouveau concept, à savoir le problème de la planification généralisée des trajets pour la gestion du trafic des UAS. Dans ce problème, les trajectoires des UAS sont définies par les opérateurs à l'aide de séquences de groupes de points de cheminement, et la trajectoire résultante doit traverser au moins un point de cheminement de chaque groupe. Ce problème est particulièrement pertinent pour les applications impliquant de multiples stations de recharge et des lieux de ramassage/dépose répartis dans une zone de vol désignée. Leur solution génère efficacement des trajectoires sans collision, ce qui permet un CDR avant le vol. Les résultats de la simulation démontrent l'efficacité et l'extensibilité de leur méthode, même lorsque le nombre de demandes et la taille du

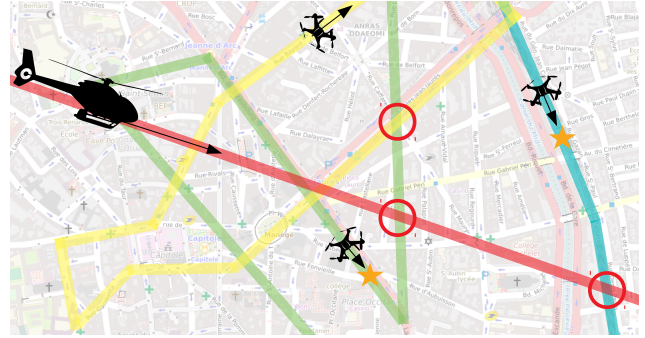


FIGURE 2 – 3 drones suivant leurs trajectoires (vert, bleu et jaune) gérant quelques incidents (étoiles orange), un hélicoptère médical sur sa trajectoire d'urgence (rouge), et des conflits identifiés (cercles rouges), d'après [17].

graphe augmentent.

[18] explorent les stratégies de gestion des conflits pour les drones autonomes. Ils examinent les procédures de déconfliction qui évitent les manœuvres horizontales et étudient deux types de structures d'espace aérien : monocouche et multicouche. Pour évaluer l'efficacité des différents paradigmes de déconfliction, ils simulent une journée d'opérations de drones dans une zone géographique. Leurs principales contributions comprennent l'introduction de nouveaux schémas de déconfliction pour les drones et une évaluation des conceptions de l'espace aérien à une ou plusieurs couches. Leurs conclusions indiquent qu'une approche centralisée ne permet pas toujours d'obtenir les meilleures performances.

[17] utilise des méthodes de résolution DCOP pour coordonner la décision des actions à sélectionner pour réparer les contrats 4D conflictuels. Pour chaque conflit, l'action qui minimise le nombre de conflits, le nombre de points de passage manqués et le retard qui en résulte est sélectionnée. Bien que cette approche ait montré de bonnes performances par rapport à un solveur centralisé, la solution ne prend pas en compte l'énergie des drones (alors qu'il s'agit d'une information cruciale lorsqu'il s'agit de véhicules ayant une autonomie énergétique très limitée) ni les décisions collectives multicritères et guidées par les concessions (certains drones peuvent devoir effectuer des actions et donc perdre en qualité de service, alors que d'autres n'ont pas fait de concession sur leurs contrats).

3 Scénario illustratif

Nous nous inspirons du document ConOps de la FAA [6] et du scénario utilisé dans [17], comme le montre la figure 2. Il est nécessaire de réviser les contrats des drones afin d'éviter les trajectoires contradictoires. Les trajectoires sont demandées à l'UTM, qui fournit alors les points de repère 4D que les drones doivent suivre pour accomplir leurs tâches, ainsi que des marges de tolérance présentées comme des tubes de sécurité. Les trajectoires initiales fournies par l'UTM sont garanties sans conflits spatiaux et temporels, mais doivent être demandées plusieurs heures avant le début des opérations.

Cependant, dans certaines situations, les drones peuvent rencontrer des difficultés à respecter leur trajectoire désignée.

Par exemple, des vents imprévisibles peuvent modifier considérablement ces trajectoires, en particulier pour les drones petits et légers [16]. En outre, des tâches spécifiques peuvent nécessiter qu'un drone maintienne sa position afin d'exécuter des routines de surveillance. Enfin, des trajectoires hautement prioritaires peuvent être introduites dans l'espace aérien en cas d'urgence. Les contrats 4D ne gèrent pas ces trajectoires conditionnelles et non déterministes, et doivent donc être mis à jour lorsque de tels événements se produisent.

Nous considérons ici les drones ayant des trajectoires de surveillance prédéfinies, qui peuvent avoir besoin de rester en vol stationnaire pour observer une scène imprévisible, interrompant ainsi temporairement leur trajectoire existante, comme dans [17]. Nous examinerons également certains événements d'urgence, dont la destination n'est pas non plus connue à l'avance, qui sont hautement prioritaires et ne peuvent être négociés, comme dans [6]. Ces deux types d'événements généreront très probablement des conflits entre les trajectoires. Ce scénario s'inscrit dans la couche UAS de la figure 1, où les drones peuvent s'engager dans un échange d'informations direct par le biais de la communication V2V, recevoir des commandes directives de la part des opérateurs et accéder à des informations en temps réel (par exemple sur les trajectoires, les conflits, les procédures d'urgence) de la part de certains USSP. Les mécanismes de coordination proposés seront mis en œuvre par plusieurs drones (semi-)autonomes ou opérateurs UAS, en fonction de l'autonomie des véhicules.

4 Concepts et modèle du problème

Cette section présente les modèles de base utilisés dans cette étude, à savoir les trajectoires 4D, les drones et les services UTM nécessaires pour mettre en œuvre les actions de déconfliction entre les drones. Nous réutilisons principalement les concepts et les notations de [17].

4.1 Trajectoires 4D

Définition 1. Une trajectoire ω est un ensemble $W \subset \mathbb{R}^4$ de 4D points $w = (x, y, z, t)$ où x et y sont des coordonnées sur le plan 2D (ou des coordonnées GPS), z est l'altitude, et t le temps.

Définition 2. Un segment $l = (w_s, w_e)$ est une ligne entre deux points consécutifs opportuns de la même trajectoire.

Seuls les segments horizontaux et verticaux sont autorisés. Les drones volent sur des plans en suivant des segments horizontaux ou d'un plan à l'autre en utilisant des segments verticaux. L'espace aérien est divisé en plusieurs plans séparés par une hauteur constante, notée z_{sep} . Sur chaque plan, les routes sont libres, c'est-à-dire qu'il n'y a pas de réseau structuré prédéfini.

Définition 3. Les tubes de sécurité sont des volumes définis horizontalement (pour les dimensions x et y), verticalement (pour z) et temporellement (pour t), par $\tau = (h, v, t)$, autour de chaque segment d'une trajectoire ω , notée $\sigma_\tau(\omega)$.

Définition 4. La trajectoire ω_1 est en conflit avec la trajectoire ω_2 lorsqu'il existe au moins un point des segments de ω_1 à l'intérieur du tube de sécurité de $\sigma_\tau(\omega_2)$.

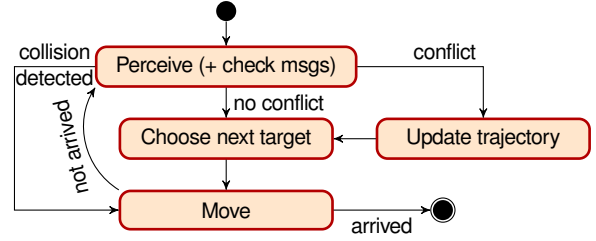


FIGURE 3 – Le modèle générique de drone.

Nous redirigeons vers [17, p.3] pour les conditions d'inclusion d'un point à l'intérieur d'un tube de sécurité. Notez que lorsque $\tau = (h, v, t)$ est le même pour tout l'espace aérien, si ω_1 est en conflit avec $\sigma_\tau(\omega_2)$, alors ω_2 est en conflit avec $\sigma_\tau(\omega_1)$.

4.2 Modèle de drones

Notre scénario considère un ensemble U de drones, en ajoutant l'état de charge au modèle de [17]. Il est primordial de surveiller attentivement l'état de charge de la batterie pendant les opérations des drones pour assurer la sécurité, la réussite de la mission et la préservation de l'équipement. Maximiser le niveau de la batterie garantit la sécurité des vols et le respect des réglementations aéronautiques [5]. Elle permet également d'optimiser l'efficacité des missions futures et l'intégrité des données en évitant les arrêts inopinés en cours de vol.

Définition 5. Un drone est défini par un tuple $u = (p, s, d, c, \omega)$, où $p = (x, y, z, t) \in \mathbb{R}^4$ est sa position, $s = (h, v, a) \leq (h_{max}, v_{max}, a_{max}) \in \mathbb{R}^3$ représente sa vitesse horizontale actuelle (en $m.s^{-1}$), verticale (en $m.s^{-1}$) et angulaire ($rad.s^{-1}$). $d \in [0, 2\pi]$ définit sa direction actuelle, c est son état de charge actuel et ω est sa trajectoire/contrat 4D.

Dans des conditions nominales, les drones changent de vitesse et de direction afin d'atteindre leur prochain point de cheminement à temps ou de retourner dans leur tube de sécurité s'ils ont été forcés d'en sortir (par exemple, en raison d'un vent fort ou pour éviter un obstacle). Lorsqu'il y a une scène à observer, un drone reste à la même position (il garde un point de cheminement à sa position actuelle) pour observer la situation pendant un temps donné, en effectuant un vol stationnaire. Il reprend ensuite sa trajectoire nominale. [17] modélise ce comportement de suivi de cible comme un modèle de mouvement simple, non détaillé ici, mais nécessaire pour simuler et évaluer les conséquences des actions correctives. En utilisant ce comportement de suivi de cible, les drones suivent une boucle classique percevoir-décider-agir, décrite dans la figure 3. Les drones interagissent avec les services UTM (par exemple, géolocalisation, détection des conflits, révision des contrats 4D) et avec les drones voisins (soit par V2V, soit par USSP).

Il est important d'ajouter la notion d'état de charge c . Lorsqu'un drone est en dessous d'un seuil d'énergie donné, il retire sa trajectoire pour atterrir en toute sécurité à la position sûre la plus proche [22]. Nous utilisons un modèle énergétique simple, qui consiste en une fonction de consommation définie comme suit :

Définition 6. La consommation d'énergie pour une vitesse 3D donnée et une durée dt , est donnée par

$$E = (c_f + c_v \max(0, v) + c_h (h^2)) dt \quad (1)$$

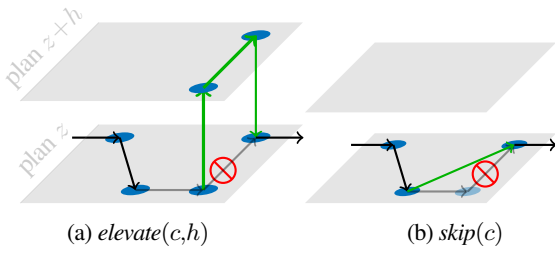


FIGURE 4 – Actions correctives ajoutant et supprimant des points de passage (cercle) et des segments (flèches) pour résoudre le conflit c (signe rouge) : les points de passage et les segments retirés sont transparents, les points de passage et les segments ajoutés sont en vert.

où le paramètre c_f est la consommation d'énergie fixe pour l'activité non inertielle du drone (calcul, communication, vol stationnaire, etc.), le paramètre c_v (resp. c_h) correspond à l'énergie pour les déplacements verticaux (resp. horizontaux).

Notez que la composante de vitesse angulaire a est considérée comme négligeable par rapport à c_f , et n'est pas utilisée pour évaluer la consommation d'énergie. Ces paramètres peuvent être déduits des caractéristiques des drones (puissance, poids, etc.). Ce modèle énergétique sera utilisé pour (i) les simulations utilisées dans l'évaluation expérimentale, où les vecteurs de vitesse sont les vitesses réelles des drones, et (ii) les évaluations des mesures correctives, où la vitesse est fixée à la vitesse de croisière des trajectoires.

4.3 Actions correctives

Pour répondre aux événements imprévus, les trajectoires doivent être corrigées afin d'éviter les conflits et donc les collisions. Avec les contrats 4D, l'espace d'action est immense et la résolution des conflits peut être complexe et difficile à comprendre. Par conséquent, comme dans la gestion traditionnelle du trafic aérien, nous donnons la priorité à des actions simples mais efficaces. De plus, la structure de l'espace aérien que nous considérons consiste en des plans séparés par une distance prédéterminée, les conflits consistent en des intersections sur le même plan. De plus, les drones sont capables d'effectuer des vols stationnaires. Compte tenu de ces contraintes opérationnelles, trois mesures correctives ont été identifiées : *postpone*, *elevate* et *skip*. [17].

Ces corrections simples favorisent l'évolutivité, la clarté et la prévisibilité : *postpone*(c,d) consiste à retarder tous les points de passage du contrat après une date donnée pour une durée donnée d , de manière à pouvoir éviter un conflit c , sans modifier la structure spatiale du graphe ; *elevate*(c,h) consiste à ajouter des points de passage pour changer de plan (soit vers le bas, soit vers le haut de la distance h), de manière à contourner l'emplacement du conflit c , puis à revenir à son plan initial ; *skip*(c) retire le segment sur lequel se produit le conflit c , en supprimant son point de passage de départ. Les changements structurels résultant de *elevate* et *skip* sont illustrés dans la Figure 4.

Pour réparer un ensemble de trajectoires conflictuelles, des actions correctives séquencées peuvent s'avérer nécessaires, car toute action corrective peut générer des conflits entre segments

après l'instant du conflit.

Proposition 1. *Il existe toujours une séquence de postpone actions correctives, avec une durée fixe d , qui résout tous les conflits pour un ensemble de trajectoires, dans l'hypothèse d'une énergie illimitée.*

Démonstration. Nous le prouvons en générant une séquence d'actions *postpone* d'une durée d , de sorte que tous les drones, à l'exception d'un drone u , reportent suffisamment longtemps pour que la trajectoire de u atteigne sa destination sans conflit (il faut autant de postpone que le temps restant pour que u atteigne sa destination divisé par d). Nous répétons le processus jusqu'à ce que tous les drones aient été traités. Cela revient à séquencer toutes les trajectoires, l'une après l'autre. \square

Toutefois, cela n'est vrai que dans le cas d'une énergie illimitée, car les trajectoires résultantes peuvent nécessiter plus d'énergie que celle qui est disponible. Une telle situation, dans laquelle tous les drones sauf un doivent attendre, n'est pas optimale, et même si elle garantit l'arrêt du processus, dans la pratique, de nombreuses autres séquences peuvent aboutir à des situations sans conflit. Dans le présent document, nous tiendrons compte des ressources énergétiques limitées, tout en supposant que les drones ont la capacité d'atterrir en toute sécurité à partir de leur position actuelle, si nécessaire [22].

Problème (4D-CRP). *Étant donné un ensemble de drones U , le Problème de réparation de contrats 4D (ou 4D-CRP) consiste à trouver un ensemble d'actions correctives pour résoudre tous les conflits entre les trajectoires des drones de U , tout en minimisant le coût global des actions correctives.*

La proposition 1 garantit qu'il existe au moins une solution à tout 4D-CRP. Toutefois, il est très probable que plusieurs solutions existent et que certaines soient meilleures que d'autres. Nous devons donc introduire des critères d'évaluation pour déterminer le coût des mesures correctives. En outre, ces critères devraient aider les opérateurs à comprendre les raisons qui sous-tendent la sélection des mesures correctives.

4.4 Évaluer le coût des actions correctives

Les actions correctives ont un impact différent sur les trajectoires : *postpone* ajoute du retard, *skip* retire un point de passage tout en conservant les mêmes points temporels, et *elevate* ajoute des points et du retard. Ces actions dégradent donc la qualité de service, c'est-à-dire le respect du contrat 4D initial. Mais plus important encore, du point de vue de la sécurité, ces actions peuvent générer d'autres conflits lors de la résolution de certains autres.

Nous considérons donc les fonctions suivantes pour évaluer le coût de l'action a quel que soit le drone qui l'exécute.

$\kappa_c(a)$ est la différence entre le nombre initial de conflits avant d'effectuer l'action a et les conflits dans l'ensemble de trajectoires résultant.

$\kappa_b(a)$ est la consommation d'énergie résultant de l'exécution de l'action a , ce qui nécessite d'évaluer l'énergie supplémentaire requise pour exécuter la nouvelle trajectoire à l'aide de l'équation 1.

$\kappa_d(a)$ est le délai résultant de l'exécution de l'action a . Pour $postpone(c, d)$ (resp. $skip(c)$) c 'est d (resp. 0), et pour $elevate(c, h)$, c 'est le temps de vol vers le haut et vers le bas de h .

$\kappa_w(a)$ est le nombre de points de passage manqués, c'est-à-dire 1 pour $skip$, 0 sinon.

Pour mettre en œuvre une évaluation multi-objectifs, nous considérons les critères de manière lexicographique, par exemple l'ordre $\kappa_c \succ \kappa_w$ signifie que nous préférons les actions avec un minimum de conflits, puis pour un nombre égal de conflits, nous préférons les actions retirant moins de points de passage. Pour entrer dans le cadre d'optimisation de la 4D-CRP, la minimisation peut être remplacée par *optimisation lexicographique*, puis encodée dans une somme pondérée équivalente [24]. Nous utiliserons cette technique dans nos algorithmes. Notons que le coût résultant d'un processus de résolution peut ensuite être décodé en un vecteur lexicographique, à des fins d'explication. Dans [17], seul l'ordre $\kappa_c \succ \kappa_w \succ \kappa_d$ est pris en compte, ce qui est limité pour deux raisons. Premièrement, l'énergie n'est pas prise en compte dans l'évaluation de l'action, alors qu'il pourrait s'agir d'un critère déterminant pour décider quel drone doit agir et donc consommer de l'énergie. Cette limitation peut être résolue en utilisant le critère κ_b . Deuxièmement, les actions passées des drones ne sont pas prises en compte lors de l'évaluation des actions. Par conséquent, il peut arriver que certains drones effectuent plus d'actions que d'autres, ce qui conduit à des situations injustes. Pour pallier cette limitation, nous proposons d'utiliser des critères liés aux concessions passées :

$\overline{\kappa_b}(u)$ est l'énergie totale concédée au cours des actions correctives passées menées par u .

$\overline{\kappa_d}(u)$ est le retard total concédé au cours des actions correctives menées par u .

$\overline{\kappa_w}(u)$ est le nombre total de points de passage retirés au cours des actions correctives passées menées par u .

Pour garantir la sécurité, nous considérerons dans nos expériences les critères lexicographiques avec κ_c comme priorité absolue ($\kappa_c \succ \kappa$ pour tout $\kappa \neq \kappa_c$).

5 Méthodes de résolution

Nous présentons trois algorithmes que nous avons mis en œuvre pour résoudre le 4D-CRP. Nous proposons des algorithmes de sélection d'actions séquentielles, afin de sélectionner des actions correctives de manière réactive. De plus, en raison des conséquences spatio-temporelles des actions, une approche pertinente consiste à considérer les conflits dans un ordre chronologique, ce qui correspond à la nécessité pour les actions correctives d'être compréhensibles pour les opérateurs de surveillance humains.

5.1 Recherche par parcours de graphe

La détermination de l'ensemble des actions à effectuer pour résoudre la 4D-CRP peut être modélisée comme un problème de recherche [19], où : (i) Chaque état $s = \langle \Omega, \hat{\kappa} \rangle$ est composé d'un ensemble de trajectoires $\Omega = \{\omega, \forall u = (p, s, d, c, \omega) \in U\}$, et d'une cartographie $\hat{\kappa} : u \mapsto \overline{\kappa}(u)$ associant à chaque drone u dans U un certain coût d'évaluation évalué par les critères $\overline{\kappa}$, qui peuvent être n'importe quel critère spécifique à u (comme

$\overline{\kappa_b}$, $\overline{\kappa_d}$ ou $\overline{\kappa_w}$) ou n'importe quelle combinaison lexicographique de ces critères ; (ii) Un état initial s_0 est constitué des trajectoires actuelles et des concessions actuelles faites par chaque drone u jusqu'à présent ; (iii) Un état but est tout état s où Ω est sans conflit - dont l'existence est garantie par la Proposition 1 ; (iv) La fonction *successeur* consiste à effectuer une action a pour résoudre le premier conflit (dans l'ordre chronologique) dans l'état $s = \langle \Omega, \hat{\kappa} \rangle$ qui résulte dans l'état $s' = \langle \Omega', \hat{\kappa}' \rangle$, où Ω' est le nouvel ensemble de trajectoires (seule une trajectoire ω a été mise à jour), et $\hat{\kappa}'$ est mis à jour avec les coûts d'évaluation pour le drone u effectuant l'action a ; (v) La fonction de coût pour l'exécution de l'action a dans l'état $s = \langle \Omega, \hat{\kappa} \rangle$ peut être n'importe laquelle des fonctions d'évaluation d'action susmentionnées (voir la section précédente) calculée en utilisant soit a , soit a et $\hat{\kappa}$.

En raison de la nature cumulative du coût (même pour les fonctions lexicographiques codées en sommes pondérées équivalentes) et de la nature non informée du problème (pas de limite supérieure informative en raison d'événements imprévisibles), nous optons pour une *recherche à coût uniforme* (UCS) [19], qui explore d'abord les états résultant d'actions dont le coût cumulatif du chemin est minimal. Nous notons $ucs(P)$ le résultat de ce processus de recherche sur 4D-CRP P . Il renvoie une séquence d'actions A^P et un coût κ^P .

La solution est une séquence d'actions visant à atteindre des trajectoires sans conflit, qui peuvent mettre à jour des segments dans un avenir lointain, par rapport à la fréquence des événements générateurs de conflits. Garantir une telle absence de conflit à long terme peut s'avérer inutile ; ainsi, limiter à la fois l'horizon temporel (en ne considérant que les k premiers points de chaque trajectoire) et la profondeur de la recherche (c'est-à-dire une séquence de taille k) sont deux moyens faciles d'améliorer le temps de calcul, sans pour autant dégrader la qualité.

Il convient de noter que cette solution est entièrement centralisée, ce qui nécessite de fournir un service à l'ensemble de l'espace aérien pour installer la correction de trajectoire.

5.2 Enchères séquentielles

Considérons maintenant une approche plus distribuée, où chaque drone impliqué dans le premier conflit dans l'ordre chronologique calcule une séquence d'actions qu'il peut effectuer par lui-même sur les trajectoires conflictuelles, puis partage cette solution avec ses voisins en conflit, et enfin la séquence à coût minimal est sélectionnée pour résoudre le conflit. Cette approche locale peut s'appuyer sur la même procédure de recherche que précédemment, mais limitée aux actions d'un seul drone, et est notée $ucs(P, u)$ pour le drone u , et renvoie une séquence d'actions et son coût, et donc son vecteur lexicographique.

Cette approche, illustrée dans l'algorithme 1, revient à mettre en œuvre des enchères séquentielles à un seul élément (SSI), classiquement utilisées dans l'attribution des tâches [20]. Ici, les éléments sont des conflits et les offres sont évaluées en résolvant le 4D-CRP P en limitant les actions à un seul drone u , en appelant $ucs(P, u)$. Une fois qu'un gagnant est déterminé pour un conflit donné, la séquence d'actions gagnante est exécutée sur les trajectoires Ω , en appelant $apply(a, \Omega)$. Bien qu'elle ne soit pas entièrement coordonnée, cette approche locale présente l'avantage de permettre à chaque drone de calculer une séquence

Algorithme 1 : Enchères séquentielles

Data : Un 4D-CRP P , avec drones dans U et trajectoires Ω **Result** : une séquence d'actions $A = \{a_1, \dots, a_n\}$

```
1  $A \leftarrow \emptyset$ 
2 while  $\Omega$  a des conflits do
3    $c \leftarrow$  premier conflit par ordre chronologique
4    $U^c \leftarrow \{u \text{ impliqué dans } c\}$ 
5   for each  $u \in U^c$  do concurrently
6      $A_{u, \kappa_u^c}^c \leftarrow \text{ucs}(P, u)$  // miser
7     // envoyer  $A_{u, \kappa_u^c}^c$  au commissaire
8    $w \leftarrow \text{argmin}_{u \in U^c} \{\kappa_u^c\}$ 
9   // détermination du gagnant
10   $A \leftarrow A \cup A_w^c$ 
11  for each  $a \in A_w^c$  do  $\Omega \leftarrow \text{apply}(a, \Omega)$ 
12  // mise à jour des contrats
13 return  $A$ 
```

simple sans autorité centrale de contrôle. Cependant, un *service d'enchères* (AS) est nécessaire pour recueillir les offres et déterminer les gagnants. Ce service peut potentiellement être fourni par un drone, un USS ou être décentralisé en utilisant des approches basées sur le consensus, comme CBBA [4].

5.3 Optimisation sous contraintes distribuée

Nous considérons ici une autre approche distribuée, où le 4D-CRP est codé comme un problème d'optimisation de contraintes distribuée (DCOP), comme inspiré par [17].

Nous rappelons brièvement qu'un DCOP est un tuple $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{C}, \mu, f \rangle$. Ici, $\mathcal{X} = \{x_{u,a,c}\}$ sont des variables binaires concernant le drone u qui effectue ou non l'action a pour résoudre le conflit c ; $\mathcal{D} = \{\mathcal{D}_{x_{u,a,c}} = \{0,1\}\}$ est un ensemble de domaines binaires, un domaine par variable; L'élément \mathcal{C} est un ensemble de contraintes douces appartenant à deux catégories, à savoir les coûts unaires des actions et les contraintes "au plus un" empêchant le même conflit d'être résolu plus d'une fois. $\alpha: \mathcal{X} \rightarrow \mathcal{A}$ associe chaque variable $x_{u,a,c}$ à l'agent u ; L'élément f est une fonction objective, représentant le coût global (somme) d'une affectation de variable complète.

Nous renvoyons le lecteur à l'article [17], pour un modèle DCOP détaillé. Il est important de noter que notre modèle diffère à deux égards. Tout d'abord, nous ne considérons qu'un seul conflit à résoudre à la fois de manière itérée, alors que [17] considère tous les conflits en cours en même temps. Cela signifie que, comme pour la méthode basée sur les enchères, les agents se coordonnent séquentiellement pour résoudre un seul conflit avant de résoudre le suivant. Comme pour les deux méthodes précédentes, ceci est motivé par le fait que la résolution des conflits dans un ordre chronologique est plus efficace que la résolution des conflits à long terme avec un meilleur coût, mais qui peut être inutile lors de la résolution des conflits à court terme, tout en conduisant à des solutions plus pratiques et explicites. Deuxièmement, [17] n'a considéré qu'une seule fonction objective, équivalente à l'optimisation lexicographique de $\kappa_c \succ \kappa_w \succ \kappa_d$. Ici, nous évaluerons les performances d'une formulation DCOP sur plusieurs fonctions objectives (voir expériences). Nous tirerons également parti de l'étude réalisée

dans [17], en utilisant uniquement l'algorithme Asynchrone Forward Bounding (AFB) [8], qui a affiché les meilleures performances parmi une grande famille d'algorithmes DCOP.

6 Évaluation expérimentale

Nous comparons ici les algorithmes proposés, combinés à des critères d'évaluation, sur des ensembles de trajectoires perturbées par des incidents et des trajectoires d'urgence.

Les expériences sont mises en œuvre en Java et exécutées sur un processeur Intel(R) Xeon(R) E5-2660 v3 à 20 cœurs @ 2,60 GHz, 62 Go de RAM, Ubuntu 18.04.5 LTS, avec une JVM OpenJDK 11.0.9. Nous exécutons 30 instances d'ensembles de trajectoires et d'incidents générés aléatoirement pour chaque taille de flotte $n = \{5, 10, 15, 20, 25\}$, et nous traçons les valeurs moyennes, avec un intervalle de confiance de $[0.05, 0.95]$. Le solveur DCOP utilisé est l'implémentation AFB de la bibliothèque FRODO [11]. Les valeurs générées aléatoirement sont choisies uniformément dans les intervalles fournis. Le temps de calcul indiqué est le temps de simulation mono-CPU.

Nous rappelons les paramètres de génération du scénario de [17]. La zone est de 1000m par 1000m, avec des plans d'espace aérien verticaux à 20m, 40m et 60m. $h_{max} = 18m.s^{-1}$, $v_{max} = 6m.s^{-1}$, $a_{max} = \Pi/2rad.s^{-1}$, $\Delta h_{max} = \Delta v_{max} = 6m.s^{-2}$, $\Delta a_{max} = \Pi/2rad.s^{-2}$. La vitesse initiale est fixée à $(0,0,0)$. Les trajectoires initiales du drone sont générées de manière aléatoire et incrémentale en choisissant au hasard 30 points séparés par une distance de 200m à 1000m, éventuellement sur des plans séparés. Les trajectoires ont un segment supplémentaire pour revenir à leur point d'origine. Les tubes de sécurité sont définis par $(h,v,t) = (30m, 15m, 10s)$. Des trajectoires d'urgence de 10 sont générées, partant d'une position aléatoire à la frontière de la zone, se déplaçant vers un point aléatoire à une distance d'au moins 1km du point initial, puis un point d'attente pour 360s, et enfin un chemin de retour vers son origine. Des incidents supplémentaires se produisent avec 5% de chance toutes les secondes, à proximité d'un drone choisi aléatoirement avec une distribution normale $\mathcal{N}(0.5, 0.15)$ mise à l'échelle et coupée en $[0, n]$ pour déséquilibrer la charge entre les drones devant effectuer un vol stationnaire pendant 30s à 120s. Nous utilisons le même modèle énergétique que celui défini par l'équation 1, et équipons les drones de batteries de 30000mAh (provenant de missions de plus d'une heure).

Nous évaluons : **ucs**, qui résout les conflits avec le solveur centralisé basé sur la recherche graphique; **ssi**, qui résout les conflits avec les enchères séquentielles d'un seul article; **sdcop**, qui résout les conflits (un par un) avec AFB. Les drones peuvent effectuer les actions suivantes : *postpone*(c, d) avec $d \in \{20, 40, 60\}$, *elevate*($c, \pm 20$), et *skip*(c). Ces actions sont évaluées à l'aide de certains critères lexicographiques, qui ont tous κ_c en premier (pour garantir la sécurité), et utilisent toujours le hasard pour départager les ex-aequo. Dans la figure, nous les notons comme suit : **b** $\equiv \kappa_c \succ \kappa_b$, **d** $\equiv \kappa_c \succ \kappa_d$, **w** $\equiv \kappa_c \succ \kappa_w$, **wd** $\equiv \kappa_c \succ \kappa_w \succ \kappa_d$, **bwd** $\equiv \kappa_c \succ \kappa_b \succ \kappa_w \succ \kappa_d$, **b concession** $\equiv \kappa_c \succ \overline{\kappa_b} \succ \kappa_b$, **d concession** $\equiv \kappa_c \succ \overline{\kappa_d} \succ \kappa_d$, **and w concession** $\equiv \kappa_c \succ \overline{\kappa_w} \succ \kappa_w$.

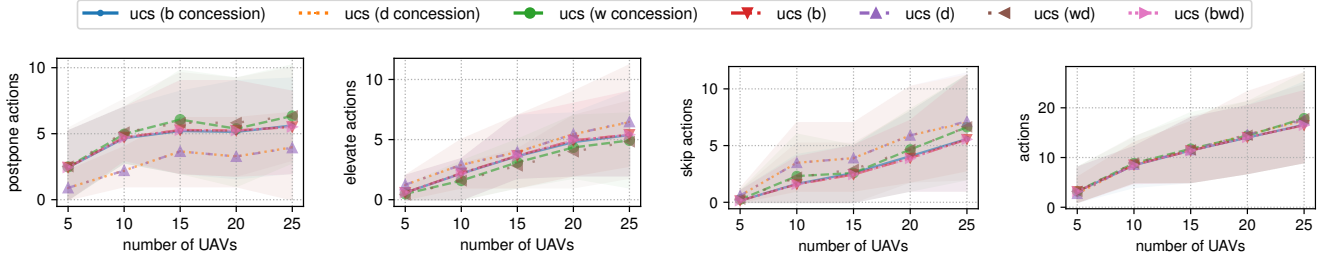


FIGURE 5 – Décisions prises avec les différents critères d'évaluation des coûts lorsqu'ils sont utilisés avec le solveur UCS.

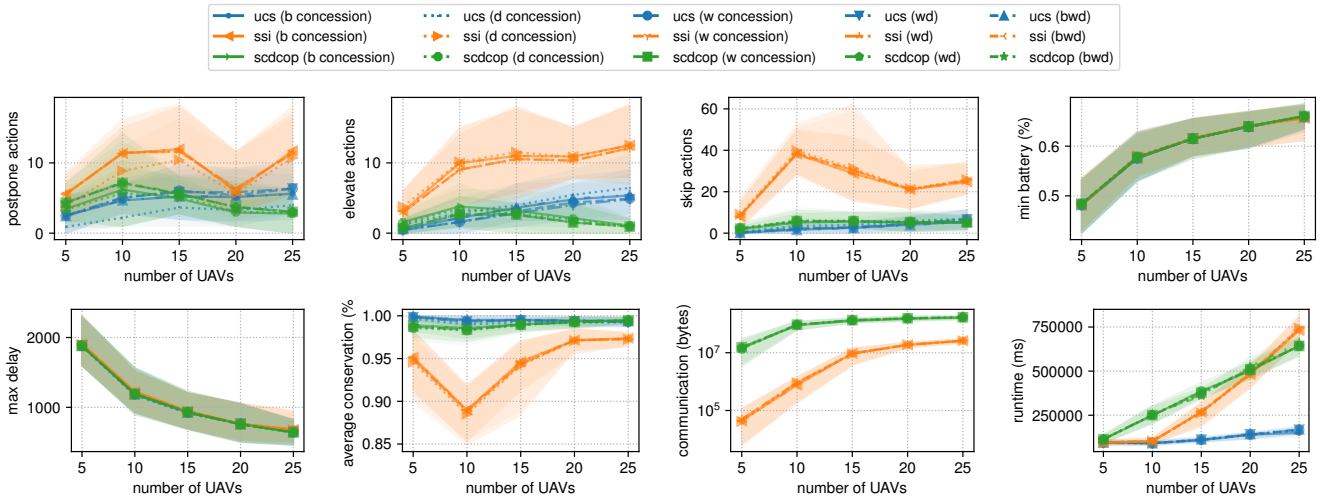


FIGURE 6 – Valeurs moyennes sur 30 instances pour plusieurs mesures de performance avec un nombre croissant de drones.

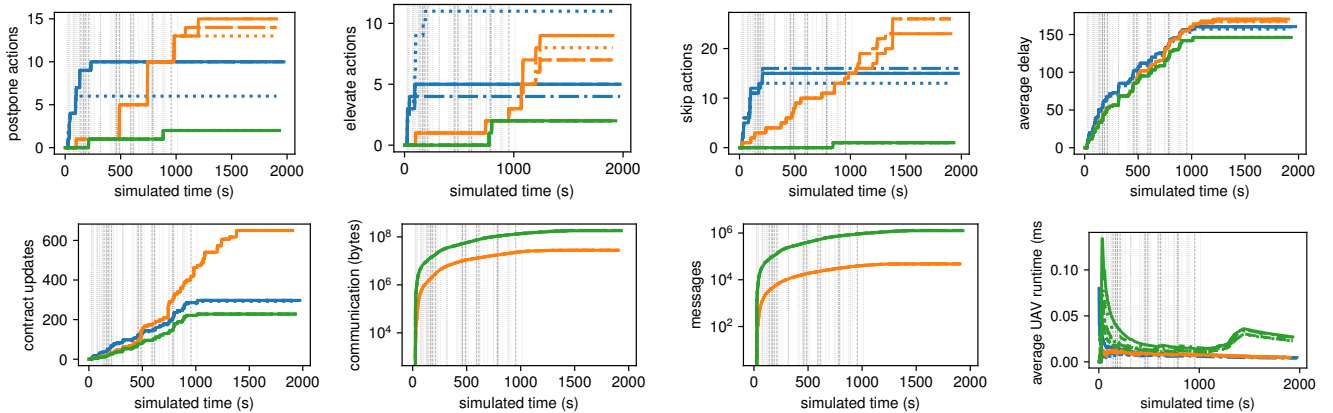


FIGURE 7 – Résultats d'une simulation avec 25 drones et 10 procédures d'urgence (en pointillés gris) et 46 incidents (en pointillés gris).

6.1 Effets des critères sur les choix d'action

Pour comprendre les conséquences de l'utilisation de certains critères d'évaluation par rapport à d'autres, nous analysons le nombre de fois où les trois actions (*postpone*, *elevate*, *skip*) sont déclenchées pour chacun de ces critères d'évaluation, comme l'illustre la Figure 5. Pour des raisons de lisibilité, nous présentons ici les résultats du solveur centralisé UCS seul. Nous pouvons observer de légères divergences dans les actions préférées résultant de l'utilisation de différents critères. Les critères d et d concession préfèrent utiliser les actions *postpone* et promouvoir les actions *skip* puis *elevate* afin de réduire les délais. w, wd et w concession favorisent *elevate* pour conserver autant de points de passage que possible. Enfin,

en ce qui concerne les critères liés à la batterie, b, bwd et b concession tendent à réaliser des compromis entre les deux familles susmentionnées en ce qui concerne *postpone* et *elevate*, et évitent étonnamment d'utiliser les actions *skip*.

Il est à noter que, malgré la densité de l'espace aérien que nous générons, relativement peu d'actions correctives sont déclenchées (environ 20, ce qui est encore extrême dans le domaine UTM, où la gestion de plus de 2 conflits à la fois est très improbable), quel que soit le critère d'évaluation. L'espace libre est très flexible et peu sujet aux conflits, contrairement à l'espace aérien structuré. Ainsi, malgré la forte dynamique et la fréquence élevée des événements imprévisibles, peu de conflits sont générés. Compte tenu de l'horizon court utilisé dans ces simulations, les mécanismes de concession ne ressortent pas vraiment, car les

agents n'ont pas assez de temps pour accumuler des concessions afin d'en tirer profit lors de la résolution des conflits. Cependant, nous pensons que ce mécanisme pourrait être utile dans des contextes plus longs, même avec des flottes multiples.

6.2 Comparaison des solveurs 4D-CRP

Nous allons maintenant comparer les performances des différents algorithmes dans le même contexte, illustré dans la Figure 6. Nous analysons en particulier le nombre d'actions déclenchées par type, comme dans la section précédente, dans les trois premières figures de la première ligne. Il est clair que SSI déclenche beaucoup plus d'actions correctives de tout type, et donc de nombreuses révisions du contrat 4D. Cela est dû à sa façon de résoudre chaque conflit en utilisant une séquence d'actions, à l'aide d'une version locale de UCS, qui peut s'avérer inutile à l'avenir. Bien qu'il soit très rapide dans des contextes plus restreints (figure en bas à droite), il nécessite presque 8 fois moins de partage d'informations que sdcop (deuxième figure en partant du bas à gauche). Curieusement, SSI éprouve des difficultés dans certains contextes (taille 10) : il génère de nombreuses actions *skip*. Cela signifie que SSI ne peut pas trouver de séquences d'actions mono-agent capables de sauvegarder des waypoints, en utilisant *elevate* ou *postpone* ; cela est probablement dû au fait que les agents ne peuvent pas trouver de bonnes séquences de *postpone*, puisqu'ils sont déjà tous aux prises avec l'impact du délai de surveillance, et qu'ils ne peuvent pas s'équilibrer. sdcop tend également à déclencher plus d'actions que UCS, mais dans un ordre limité par rapport à SSI. En effet, sdcop ne déclenche qu'une action par conflit détecté de manière séquentielle. Il ne génère donc pas localement une séquence d'actions potentiellement inutiles, comme le fait SSI. sdcop enregistre autant de points de passage que UCS dans des contextes plus larges. Il est intéressant de noter qu'avec des flottes plus importantes, le temps d'exécution de sdcop augmente linéairement par rapport à SSI et UCS, étant donné qu'il ne prend en compte que les séquences d'actions uniques. Enfin, il est particulièrement intéressant de constater que tous les algorithmes améliorent l'équité des délais et des batteries : plus il y a d'agents, plus il y a d'actions permettant d'équilibrer l'allocation des délais (délai maximum) et les dépenses d'énergie (niveau minimum de batterie) face à un événement (alors que ces deux éléments sont proportionnels au nombre d'agents).

Enfin, la figure 7 montre l'évolution de certaines mesures au cours de l'exécution d'un scénario. Il est intéressant de noter que UCS répare principalement les conflits au début du scénario, ce qui se traduit par des trajectoires plus robustes aux conflits à venir ; tandis que sdcop déclenche peu d'actions tout au long de la ligne temporelle. Les séquences de SSI sont souvent révisées jusqu'à la fin du scénario. Cependant, même les décisions mono action (pourtant très pratiques à appréhender du point de vue de l'opérateur) requièrent une grande quantité de données et de messages. Notez que la bosse à la fin de la figure en bas à droite (durée d'exécution moyenne des drones) est due au fait que certains drones sont arrivés et ne sont pas comptés dans la somme globale, augmentant ainsi la valeur moyenne.

7 Conclusion

L'approche de cet article consiste à résoudre le problème de réparation de contrat (4D-CRP) à chaque fois que les drones détectent des conflits. Nous avons étudié plusieurs solveurs : UCS, SSI, et sdcop. En outre, plusieurs fonctions d'évaluation du coût des actions ont été définies, sur la base des conséquences immédiates et des concessions passées. Nous avons également intégré le critère de l'énergie dans la décision, ce qui favorise les actions visant à économiser la batterie, conformément aux exigences réglementaires [5]. Nous avons évalué plusieurs combinaisons de solveurs et de critères, sur un espace aérien conflictuel. Dans l'ensemble, les opérateurs de drones et les gestionnaires d'UTM (tels que la FAA) peuvent choisir entre plusieurs mécanismes flexibles et compréhensibles pour coordonner les corrections de trajectoire, allant de la centralisation à la décentralisation. Chacun de ces mécanismes a des impacts opérationnels distincts (charge de communication, temps de calcul, changements d'altitude, etc.). En outre, les parties prenantes de l'UTM ont le choix entre plusieurs critères, afin d'améliorer l'acceptabilité et de mettre en place des décisions explicables, consistant en des séquences d'actions avec des évaluations ordonnées correspondantes. Pendant le déploiement, ces priorités pourraient être fixées par consensus social, en tenant compte des préférences de la communauté touchée, telles que la minimisation du vol stationnaire, etc.

Nos contributions ont été évaluées sur des scénarios comportant un nombre limité de conflits. L'extension à des contextes plus vastes, sur plusieurs heures, avec de nombreuses itérations de drones, pourrait mettre davantage en évidence les avantages de la prise en compte des concessions, en particulier dans les scénarios impliquant des flottes hétérogènes exploitées par différents acteurs. Cela pourrait également être le cas entre plusieurs flottes aux caractéristiques et missions hétérogènes, exploitées par des acteurs distincts. D'autres approches fondées sur le marché (et non coopératives) doivent encore être étudiées. Nous constatons que l'élaboration de longues séquences d'actions, avec une connaissance locale limitée (SSI), peut s'avérer inefficace en cas de conflits fréquents. Cependant, dans des environnements plus stables, avec une prédiction possible de la densité de l'espace aérien, de telles séquences pourraient potentiellement être efficaces, comparées aux algorithmes myopes (sdcop). En outre, le passage à des espaces aériens structurés, connus pour être plus conflictuels en raison des effets de congestion, pourrait nécessiter l'utilisation d'approches basées sur des concessions et des décisions locales, comme nous l'avons proposé. Enfin, étant donné que, dans un avenir proche, l'UTM sera géré par des opérateurs, des expériences humaines en boucle pourraient être nécessaires pour adapter les explications des mesures correctives afin d'optimiser la compréhension et la réactivité de l'utilisateur. En résumé, l'acceptabilité sociale en constante évolution, le passage changeant des algorithmes en réponse aux futures informations sur l'espace aérien urbain (par exemple, plus de prévisions sur les opérations futures) et la prise en compte des flottes avec des préférences potentielles de déconfliction interne impliquent que les algorithmes de déconfliction UTM doivent être capables d'évoluer en tandem avec l'avenir des opérations et de l'acceptation des UTM. Ces travaux constituent une première étape

dans la définition de cette catégorie algorithmique émergente.

Références

- [1] Airbus and Boeing. A new digital era of aviation : The path forward for airspace and traffic management. Technical report, Airbus, 2020.
- [2] Mehdi Bennaceur, Rémi Delmas, and Youssef Hamadi. Passenger-centric urban air mobility : Fairness trade-offs and operational efficiency. *Transportation Research Part C : Emerging Technologies*, 136 :103519, 2022.
- [3] Carlos Capitán, Héctor Pérez-León, Jesús Capitán, Ángel Castaño, and Aníbal Ollero. Unmanned aerial traffic management system architecture for u-space in-flight services. *Applied Sciences*, 11(9), 2021.
- [4] Han-Lim Choi, Luc Brunet, and Jonathan P. How. Consensus-based decentralized auctions for robust task allocation. *IEEE Trans. Robotics*, 25(4) :912–926, 2009.
- [5] European Union Aviation Safety Agency (EASA). Easy access rules for unmanned aircraft systems (regulation (eu) 2019/947 and regulation (eu) 2019/945). Technical report, European Union Aviation Safety Agency (EASA), 2022.
- [6] Federal Aviation Agency. Utm concept of operations version 2.0. Technical report, Federal Aviation Agency, 2022.
- [7] Federal Aviation Agency. Unmanned aircraft systems traffic management (utm) implementation plan. Technical report, Federal Aviation Agency, 2023.
- [8] Amir Gershman, Amnon Meisels, and Roie Zivan. Asynchronous forward-bounding for distributed constraints optimization. In *Proceedings of the 2006 Conference on ECAI 2006 : 17th European Conference on Artificial Intelligence August 29 – September 1, 2006, Riva Del Garda, Italy*, page 103–107, NLD, 2006. IOS Press.
- [9] Youssef Hamadi. Optimization for urban air mobility. In Ilias S. Kotsireas and Panos M. Pardalos, editors, *Learning and Intelligent Optimization*, pages 1–8, Cham, 2020. Springer International Publishing.
- [10] Florence Ho, Rúben Galdes, Artur Gonçalves, Bastien Rigault, Atsushi Oosedo, Marc Cavazza, and Helmut Prendinger. Pre-flight conflict detection and resolution for uav integration in shared airspace : Sendai 2030 model case. *IEEE Access*, 7 :170226–170237, 2019.
- [11] Thomas Léauté, Brammert Ottens, and Radoslaw Szymanek. FRODO 2.0 : An open-source framework for distributed constraint optimization. In *Proceedings of the IJCAI'09 Distributed Constraint Reasoning Workshop (DCR'09)*, pages 160–164, Pasadena, California, USA, July 13 2009. <https://frodo-ai.tech>.
- [12] J. Lieb and A. Volkert. Unmanned aircraft systems traffic management : A comparison on the faa utm and the european corus conops based on u-space. In *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, pages 1–6, 2020.
- [13] Tim McCarthy, Lars Pforte, and Rebekah Burke. Fundamental elements of an urban utm. *Aerospace*, 7(7), 2020.
- [14] Rafael Papa, Ionut Cardei, and Mihaela Cardei. Generalized path planning for utm systems with a space-time graph. *IEEE Open Journal of Intelligent Transportation Systems*, 3 :351–368, 2022.
- [15] Mercedes Pelegrín, Claudia D’Ambrosio, Rémi Delmas, and Youssef Hamadi. Urban air mobility : from complex tactical conflict resolution to network design and fairness insights. *Optimization Methods and Software*, pages 1–33, 08 2023.
- [16] Gabriele Perozzi, Denis Efimov, Jean-Marc Biannic, and Laurent Planckaert. Trajectory tracking for a quadrotor under wind perturbations : sliding mode control with state-dependent gains. *Journal of the Franklin Institute*, 355(12) :4809–4838, 2018.
- [17] Gauthier Picard. Trajectory coordination based on distributed constraint optimization techniques in unmanned air traffic management. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS-22)*, pages 1065–1073. IFAAMAS, 2022.
- [18] Valentin Polishchuk. Centralized and distributed utm in layered airspace. 2018.
- [19] Stuart Russell and Peter Norvig. *Artificial Intelligence : A Modern Approach*. Prentice Hall, 3 edition, 2010.
- [20] Adrian Schoenig and Maurice Pagnucco. Evaluating sequential single-item auctions for dynamic task allocation. In Jiuyong Li, editor, *AI 2010 : Advances in Artificial Intelligence*, pages 506–515, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
- [21] SESAR joint Undertaking. European atm master plan : Roadmap for the safe integration of drones into all classes of airspace. Technical report, SESAR, 2020.
- [22] Md Shah Alam and Jared Oluoch. A survey of safe landing zone detection techniques for autonomous unmanned aerial vehicles (uavs). *Expert Systems with Applications*, 179 :115091, 10 2021.
- [23] Savvy Verma, Victoria Dulchinos, Robert Dan Wood, Amir Farrahi, Richard Mogford, Megan Shyr, and Rania Ghatas. Design and analysis of corridors for uam operations. In *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, pages 1–10, 2022.
- [24] Ronald R. Yager. On the analytic representation of the leximin ordering and its application to flexible constraint propagation. *European Journal of Operational Research*, 102(1) :176–192, 1997.
- [25] Pavan Yedavalli and Jessie Mooberry. An assessment of public perception of urban air mobility (uam). *Airbus UTM : Defining Future Skies*, pages 2046738072–1580045281, 2019.