



**HAL**  
open science

## Coordination de trajectoires 4D par optimisation distribuée dans la gestion du trafic aérien sans pilote

Gauthier Picard

► **To cite this version:**

Gauthier Picard. Coordination de trajectoires 4D par optimisation distribuée dans la gestion du trafic aérien sans pilote. Conférence Nationale en Intelligence Artificielle 2022, Jun 2022, Saint-Etienne, France. pp.147-154. hal-03819952

**HAL Id: hal-03819952**

**<https://hal.science/hal-03819952>**

Submitted on 18 Oct 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.

# Coordination de trajectoires 4D par optimisation distribuée dans la gestion du trafic aérien sans pilote

Gauthier Picard  
ONERA/DTIS, Université de Toulouse, France

gauthier.picard@onera.fr

## Résumé

*Nous proposons une stratégie de déconflition dans la gestion du trafic aérien sans pilote (UTM), où les drones sont dotés de capacités de décision pour mettre à jour leurs trajectoires 4D face à des événements imprévisibles ou lorsque des trajectoires prioritaires sont ajoutées dans l'espace aérien. Les drones sont conscients des conflits potentiels proches fournis par la couche de communication UTM, et peuvent se coordonner pour réparer les trajectoires et résoudre les conflits de manière décentralisée. Cela réduit l'accès à un goulot d'étranglement décisionnel central et permet une déconflition réactive. Nous proposons des comportements coordonnés basés sur les DCOP, évalués sur des scénarios denses dans une zone limitée avec de nombreux contrats 4D, des incidents potentiels et des procédures d'urgence.*

## Mots-clés

*Gestion du trafic aérien sans pilote; Coordination; DCOP; Contrats 4D*

## 1 Introduction

En 2018, le bureau NextGen de l'administration fédérale américaine de l'aviation (FAA) a publié un premier concept d'opérations (ConOps) global pour la gestion du trafic des systèmes d'aéronefs sans pilote<sup>1</sup> (UTM) qui présente une vision et décrit les exigences opérationnelles et techniques associées pour développer une architecture support et fonctionner dans un écosystème UTM [7]. Il s'agit d'un système communautaire de gestion du trafic dans lequel les opérateurs et les entités fournissant des services de soutien aux opérations sont responsables de la coordination, de l'exécution et de la gestion des opérations, avec des règles de conduite établies par la FAA. Cet ensemble fédéré de services permet une gestion coopérative des opérations entre les exploitants de systèmes d'aéronefs sans pilote (UAS), facilitée par des fournisseurs de services tiers (USSP) grâce à des échanges d'informations en réseau. L'UTM est conçu pour répondre à la demande et aux attentes d'un large éventail d'opérations dont la complexité et le risque ne cessent de croître, grâce à un marché ouvert innovant et concurrentiel de fournisseurs de services. Les services fournis par les USSP sont interopérables pour permettre à l'écosystème UTM de répondre aux besoins de la communauté des opérateurs de drones. Ce document de travail propose également un ensemble de scénarios soulevant de fortes problématiques de coordination, dont la résolution s'avère être de véritables défis scientifiques comme l'illustre [10], et qui permettent également de définir des scénarios pour l'évaluation des solutions candidates.

<sup>1</sup> Cet article est une traduction d'un article accepté à AAMAS'22 [19].

1. Nous utiliserons dans cet article, de manière interchangeable les termes *drone* et *UAV* pour désigner des aéronefs sans pilotes.

Du côté européen, le modèle opérationnel envisagé, appelé U-Space, identifie également les besoins de coordination entre les acteurs, et de distribution des décisions [21]. Bien qu'elles présentent des différences notables (classification des drones, sectorisation de l'espace aérien, niveau de sécurité, responsabilités des USSP) comme identifié dans [13], ces ConOps convergent sur de nombreux points (orientation service, modularité, confidentialité, acteurs de haut niveau, séparation en niveaux stratégique, tactique et réactif), notamment en ce qui concerne la coordination des USSP et des drones.

Dans cet article, nous proposons un mécanisme de coordination adaptatif permettant aux systèmes d'UAV (autonomes ou opérés) évoluant dans un espace aérien urbain à route libre de gérer les conflits de trajectoire dus aux retards, aux incidents ou à l'apparition de nouvelles trajectoires hautement prioritaires. Bien qu'il existe des stratégies d'adaptation (e.g. report ou élévation de trajectoires) pour éviter les futurs conflits, nous proposons ici de faire en sorte que les drones se coordonnent en utilisant un service de messagerie directe (soit directement de véhicule à véhicule, soit en utilisant une communication à l'échelle de la plateforme) pour résoudre les conflits, et pour choisir les meilleures actions par rapport à des objectifs multiples. Le mécanisme de coordination est basé sur l'optimisation distribuée sous contraintes (DCOP), afin d'assurer que les drones choisissent collectivement des actions de déconflition, tout en minimisant certains indicateurs opérationnels. Le sujet de notre étude étant l'interaction entre les UAVs et les USSPs, les résultats proposés sont suffisamment génériques pour être applicables aux différents concepts d'opérations UTM et U-Space. Notre approche aborde la réparation de contrats 4D en ligne de manière distribuée. À notre connaissance, il n'existe pas d'approche multi-agents pour ce problème, qui est un sujet nouveau faisant suite aux récents concepts d'opération dans les espaces aériens urbains.

## 2 Scénario illustratif

Utilisons un scénario illustratif pour mettre en évidence les besoins de coordination et d'adaptation dans des contextes UTM, représentés dans la figure 1. Nous réutilisons principalement le scénario proposé dans [7], centré sur un espace aérien à route libre. Nous considérons une zone urbaine de 2 km sur 2 km où plusieurs opérateurs de drones ont demandé des trajectoires 4D pour effectuer différentes tâches, comme la livraison, la surveillance ou la capture d'images au sol. Ces trajectoires ont été demandées à l'UTM, qui en retour envoie les points de passage 4D (espace 3D et temps) que les drones doivent suivre pour accomplir leurs tâches. L'UTM fournit également des marges de tolérance permettant de s'écarter de la trajectoire initiale, représentées par des tubes de sécurité encapsulant les trajectoires. Les trajectoires (et leurs tubes de sécurité respectifs) fournies par l'UTM sont garanties sans conflit spatio-temporel mais

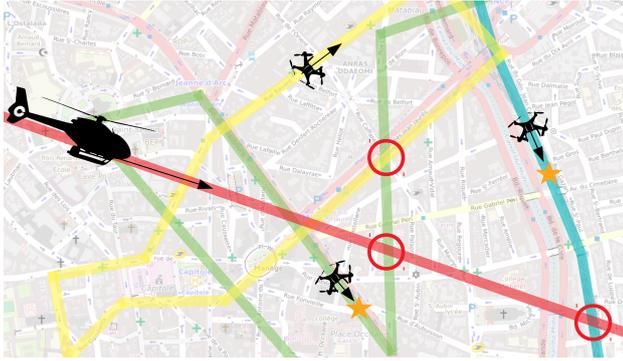


FIGURE 1 – Un scénario type avec trois drones suivant leurs trajectoires (vert, bleu et jaune) gérant quelques incidents (étoiles orange), un hélicoptère médical sur sa trajectoire d'urgence (rouge), et quelques conflits identifiés (cercles rouges)

doivent être demandées quelques heures avant les opérations. Les trajectoires 4D résultantes sont ainsi contractées (et appelées *contrats 4D*), et les drones s'engagent à respecter ces contrats.

Cependant, dans certains cas, les drones peuvent ne pas être en mesure de respecter leur trajectoire. Par exemple, un vent imprévisible peut modifier radicalement les trajectoires, en particulier pour les drones petits et légers [17]. De plus, certaines tâches peuvent exiger qu'un drone reste en position pour exécuter une routine de surveillance (e.g., un drone de surveillance détectant un danger dans sa zone de surveillance doit capturer la scène sous différents angles). Enfin, certaines trajectoires hautement prioritaires peuvent être ajoutées à l'espace aérien pour des raisons d'urgence (e.g., intervention d'un hélicoptère d'évacuation médicale), ce qui peut générer plusieurs conflits avec les trajectoires actuelles. Les contrats 4D ne gèrent pas de telles trajectoires conditionnelles et non déterministes, et doivent donc être mis à jour lorsque de tels événements se produisent.

Dans notre article, nous considérerons des drones ayant des trajectoires de surveillance, et qui peuvent avoir à mettre en œuvre une routine de surveillance d'incident (avec un vol stationnaire), suspendant ainsi leur trajectoire actuelle. L'occurrence de tels incidents n'est pas connue à l'avance, et est supposée totalement aléatoire. Nous considérerons également certains événements d'urgence, dont la destination n'est pas connue à l'avance, qui sont hautement prioritaires et ne peuvent être négociés. Dans la figure 1, trois drones suivent leurs trajectoires (verte, bleue et jaune) et gèrent certains incidents (étoiles orange), qui vont retarder leurs trajectoires planifiées, et donc générer un conflit (cercle rouge au croisement jaune-vert) à une position qui était spatio-temporellement sûre auparavant. Plus conflictuel encore, un hélicoptère médical doit traverser l'espace aérien en urgence, ce qui entraîne deux autres conflits (cercles rouges aux croisements vert-rouge et bleu-rouge).

Nous positionnons notre travail au niveau des UAS où les UAV peuvent échanger directement des informations via la communication directe, ils peuvent recevoir des commandes et des ordres des opérateurs, et obtenir des informations en temps réel (e.g. les trajectoires, les conflits, les procédures d'urgence) de certains USSP. Nous envisageons que les mécanismes de coordination tactique et réactive proposés puissent avoir lieu entre plusieurs UAS (semi-)autonomes ou entre plusieurs opérateurs UAS. Nous nous concentrons notamment sur les petits drones capables d'effectuer un vol stationnaire et opérant à basse altitude (entre 0m et 300m).

### 3 Concepts et modèle du problème

Nous présentons ici les modèles des trajectoires 4D, des drones et des services UTM nécessaires à la déconfliction.

#### 3.1 Trajectoires 4D

Une trajectoire est un ensemble  $W \subset \mathbb{R}^4$  de *points 4D*  $w = (x, y, z, t)$  où  $x$  et  $y$  sont des coordonnées sur le plan 2D (ou des coordonnées GPS),  $z$  est l'altitude, et  $t$  le temps. Ces points définissent les points de passage que les drones doivent survoler. Nous appelons un *segment*  $l = (w_s, w_e)$  une ligne entre deux points consécutifs. Nous ne considérerons que les segments horizontaux et verticaux. Les drones évoluent sur des plans ou se déplacent vers un autre plan en utilisant un segment vertical. Ainsi l'espace aérien est divisé en plusieurs plans séparés par une hauteur constante, notée  $Z_{sep}$ . Sur chaque plan, les trajectoires sont libres, c'est-à-dire qu'il n'existe pas de réseau de vol prédéfini. Une trajectoire est également définie par des marges de sécurité, afin d'éviter que les drones n'entrent en collision, laissant le temps et l'espace aux routines d'évitement réactives de subsumer les ordres de navigation. Classiquement, ces tubes de sécurité sont définis horizontalement (pour les dimensions  $x$  et  $y$ ), verticalement (pour  $z$ ) et temporellement (pour  $t$ ). Un tube de sécurité est donc défini par  $\tau = (h, v, t)$ . Un drone diverge lorsqu'il se trouve en dehors du tube de sécurité de son segment de vol actuel. Un drone viole une trajectoire si sa position est à l'intérieur du tube de sécurité d'un autre drone. Formellement, un point  $w_0$  est à l'intérieur d'un tube de sécurité  $(h, v, t)$  pour le segment  $l = (w_1, w_2)$  s'il satisfait aux trois conditions (1) et (2) et (3) :

$$\frac{|(x_2 - x_1)(y_1 - y_0) - (x_1 - x_0)(y_2 - y_1)|}{d_{xy}(w_1, w_2)} \leq h \quad (1)$$

$$\min(z_1, z_2) - v \leq z_0 \leq \max(z_1, z_2) + v \quad (2)$$

$$|t_0 - (t_1 + \frac{d_{xy}(w_1, p_{xy}(w_0, l))}{d_{xy}(w_1, w_2)} d_t(w_1, w_2))| \leq t \quad (3)$$

avec  $d_{xy}$  la distance euclidienne 2D,  $d_t$  la distance temporelle 1D, et  $p_{xy}(w, l)$  la projection 2D de  $w$  sur  $l$ . Deux segments sont en *conflit* s'ils sont sur le même plan (soit horizontalement, soit verticalement) et s'ils se coupent au même moment (à la tolérance  $t$  près).

La construction d'un ensemble de trajectoires 4D est un problème opérationnel classique qui fait l'objet d'une littérature abondante. Il a été très bien étudié dans le contexte de la gestion du trafic aérien [5]. La construction de trajectoires sans conflit est un problème d'optimisation difficile, souvent résolu à l'aide de métaheuristiques comme le recuit simulé [11] ou les algorithmes évolutionnaires [22]. En présence de petits drones capables de changer de direction et de vitesse de manière plus flexible que les avions classiques, le problème reste difficile, mais d'autres techniques comme le PSO [1] ou même des techniques multi-agents [24] ont été envisagées. Nous étudions ici un espace aérien à route libre *non structuré*, c'est-à-dire qu'il n'y a pas de réseau de routes aériennes prédéfinies auquel les trajectoires sont contraintes, contrairement aux concepts opérationnels habituels de l'ATM [16]. Dans notre cas, nous considérerons que les trajectoires initiales sont sans conflit, obtenues à partir de n'importe quel générateur de trajectoires (incrémental dans nos expériences). Nous nous concentrons sur la procédure de réparation et non sur la génération de l'ensemble initial de trajectoires. Nous cherchons à optimiser certains critères liés à la qualité de la réparation. Nous visons à : (i) *minimiser le nombre de conflits* générés par

l'adaptation de la trajectoire, afin de garantir la sécurité du vol ; (ii) *minimiser le nombre de points de passage manqués*, afin de garantir la qualité des trajectoires, notamment dans un contexte urbain, où les trajectoires sont définies pour remplir certaines missions entre et à ces points de passage ; (iii) *minimiser le retard global* induit par l'adaptation. Un tel problème n'est pas trivial et peut nécessiter un compromis ; par exemple, sauter un segment conflictuel améliore la sécurité mais réduit la qualité de service.

### 3.2 UAVs

Notre scénario considère un ensemble  $U$  de drones, chacun étant noté  $u = (p, s, d, T)$ , 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 ses vitesses horizontale (en  $m.s^{-1}$ ), verticale (en  $m.s^{-1}$ ) et angulaire (en  $rad.s^{-1}$ ) actuelles. La valeur  $d \in [0, 2\pi]$  définit sa direction actuelle, et  $T$  est sa trajectoire 4D. Notez que ces valeurs de vitesse peuvent être égales à zéro (par exemple, aucun mouvement). Dans des conditions nominales, les drones modifient leur vitesse et leur direction afin d'atteindre leur prochain point de passage à temps, ou de rentrer dans leur tube de sécurité s'ils ont été forcés d'en sortir (par exemple en raison d'un vent fort ou de l'évitement d'un obstacle). En cas d'incident à proximité de sa position, un drone restera à la même position (il garde un point de passage à sa position actuelle) pour observer la situation pendant un certain temps. Ensuite, il retournera à sa trajectoire nominale. Nous modélisons ce comportement de suivi de cible par un modèle de mouvement simple. Lorsque  $u$  est à la position  $p$  et qu'il vise à atteindre la position  $o$  avec un angle  $\angle(p, o) \in [-\Pi, \Pi]$ , il appliquera les variations suivantes à ses trois degrés de liberté :

$$\Delta h = \min\left(\frac{d_{xy}(p, o)}{d_t(p, o)}, \Delta h_{max}\right) \quad (4)$$

$$\Delta v = \min\left(\frac{d_z(p, o)}{d_t(p, o)}, \Delta v_{max}\right) \quad (5)$$

$$\Delta a = \text{sgn}(\angle(p, o)) \cdot (\min(\angle(p, o), \Delta a_{max})) \quad (6)$$

conduisant à la variation de position suivante :

$$\Delta p = (h \cdot \cos(d), h \cdot \sin(d), v, a) \quad (7)$$

Le comportement nominal d'un drone représenté sur la figure 2 consiste d'abord à percevoir l'environnement et à vérifier les messages entrants. Ensuite, si aucun conflit, ni obstacle proche n'est détecté, le drone choisit une position cible (soit un point de passage, une localisation d'incident, ou sa position contractée s'il a divergé), puis se déplace en ajustant ses vitesses à l'aide de (4), (5) et (6) pour atteindre une nouvelle position  $p + \Delta p$  le rapprochant de sa position cible. Les drones communiquent avec les services de l'UTM (par exemple, géolocalisation, détection de conflits) et avec les drones voisins (directement ou via l'USSP). Ces communications peuvent signaler des conflits. Dans ce cas, le drone déconflictera en mettant éventuellement à jour sa trajectoire. Les drones sont également équipés d'une routine réactive d'évitement des collisions s'ils détectent un mobile trop proche de leur position. Dans ce cas, le drone effectue un comportement de "détection et évitement" que nous ne détaillons pas dans cet article.

### 3.3 Services UTM

Les opérateurs de drones s'abonnent à des services afin d'obtenir des informations et de fournir des informations sur leurs missions en cours. Des informations en temps réel (sur les positions, les conflits ou les nouvelles trajectoires insérées) sont envoyées par les services aux drones abonnés, tandis que les demandes de mise à jour de trajectoire et les rapports d'incidents sont

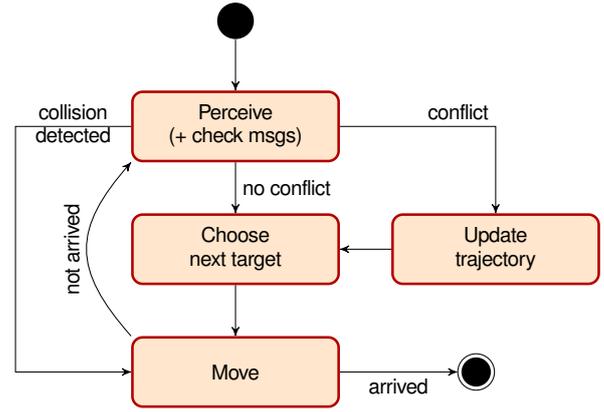


FIGURE 2 – Comportement générique d'un UAV

envoyés par les drones aux USSP. Nous considérons au moins les services suivants :

Un *service de messagerie directe* (DMS), qui permet aux UAV de communiquer entre eux. Les UAV utiliseront ce service pour envoyer des messages de coordination (liés à la méthode de résolution DCOP, voir la section 4).

Un *service de détection de conflit* (CDS), qui envoie aux abonnés les conflits devant se produire dans les prochaines minutes, suite à une mise à jour de trajectoire ou à un ajout de trajectoire. Une information de conflit contient les segments qui se croisent, les drones respectifs et le point d'intersection. On note  $C(u)$  l'ensemble des conflits dans lesquels  $u$  est impliqué. Afin de rester suffisamment réactif, la détection des conflits est effectuée à un horizon limité : elle ne vérifie que les  $k$  prochains segments pour chaque trajectoire, à partir de l'instant courant.

Un *service de mise à jour de trajectoire* (TUS), que les drones sollicitent pour reporter leur trajectoire actuelle ou pour ajouter de nouveaux points de passage (e.g. pour contourner un obstacle en dehors des tubes de sécurité). En retour, le service renvoie la nouvelle trajectoire et les nouveaux conflits (s'il y en a). En effet, si un drone doit reporter sa trajectoire, il peut produire des conflits sur les segments futurs.

Un *service d'évaluation de trajectoire* (TAS) qui est interrogé pour évaluer le remplacement d'une trajectoire par une autre. Ce service renvoie la nouvelle trajectoire et certaines mesures d'évaluation (e.g., le nombre de conflits générés et le retard résultant par rapport à la trajectoire remplacée).

### 3.4 Actions de déconfliction et comportements

Après le démarrage du système (sans conflit), des événements non prévus (comme des incidents ou des urgences) exigeront éventuellement la rupture des contrats. Puisque les conflits consistent principalement en des intersections sur le même plan, et puisque les drones considérés sont capables d'effectuer un vol stationnaire, trois options principales sont ouvertes pour mettre à jour les contrats : *postpone* (retarder), *elevate* (changer d'altitude) et *skip* (supprimer un point de passage).

**Postpone.** Son principe est simple : reporter tous les points de passage du contrat après une date donnée pour un temps donné, afin de pouvoir éviter un conflit ou d'avoir suffisamment de temps pour observer un incident. Cette action est automatiquement appelée lorsqu'un drone détecte un incident à proximité de sa position. Il demandera alors au service de mise à jour du contrat un report pour un temps dépendant de l'incident. Tous les points de passage après sa position spatio-temporelle actuelle sont alors retardés. Les drones peuvent également demander un report afin d'éviter un futur conflit potentiel identifié par

le service de détection des conflits. Dans ce cas, les drones ajouteront un nouveau point de passage avant l'intersection et retarderont leur route d'une durée égale au temps nécessaire pour sortir d'un tube de sécurité. Nous appellerons *postpone*, les UAVs effectuant uniquement cette action de déconflition.

**Elevate.** Une autre approche de la déconflition avec des drones capables de voler verticalement consiste à faire en sorte que les drones changent de plan (vers le bas ou vers le haut), afin de contourner le lieu du conflit, puis de revenir à leur plan précédent. Cette action est effectuée pour éviter les obstacles ou pour prévenir un futur conflit potentiel. Lorsqu'il demande une mise à jour de la trajectoire, le drone fournit le conflit ou la position à éviter, et le service de mise à jour de la trajectoire renvoie une trajectoire mise à jour avec un "pont" au-dessus de la position à éviter, c'est-à-dire l'ajout de 4 points de passage pour créer un segment vertical, suivi d'un segment horizontal et enfin d'un autre vertical. Un pont est soit ascendant, soit descendant (en fonction de l'altitude). Nous appellerons *elevate*, les UAV effectuant uniquement cette action de déconflition.

**Skip.** Enfin, on peut simplement sauter un segment conflictuel, en supprimant son point de départ. Ainsi, le drone contournera le conflit en se déplaçant directement vers le point de passage après le segment conflictuel. Lorsqu'il demande à sauter un segment, le drone fournit simplement les conflits au service de mise à jour de la trajectoire qui supprimera le point de passage correspondant et renverra la nouvelle trajectoire. Nous appellerons *skip*, les UAVs effectuant uniquement cette action de déconflition.

## 4 Coordination basée sur les DCOP

Cette section définit une coordination coopérative entre les drones, de sorte qu'ils décident conjointement quels drones déclencheront quelle action de déconflition pour un conflit donné. Nous modélisons ce problème de décision collective comme un DCOP.

### 4.1 Le cadre des DCOP

Un problème d'optimisation avec contraintes distribuées (DCOP) est la variante distribuée de l'optimisation avec contraintes [8]. Ici, un groupe d'agents doit choisir des valeurs pour un ensemble de variables de manière décentralisée afin de minimiser une fonction de coût. Formellement, un DCOP est un tuple  $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{C}, \mu, f \rangle$ , où :  $\mathcal{A} = \{a_1, \dots, a_{|\mathcal{A}|}\}$  est un ensemble d'agents ;  $\mathcal{X} = \{x_1, \dots, x_n\}$  sont des variables appartenant aux agents ;  $\mathcal{D} = \{\mathcal{D}_{x_1}, \dots, \mathcal{D}_{x_n}\}$  est un ensemble de domaines finis, tels que la variable  $x_i$  prend des valeurs dans  $\mathcal{D}_{x_i} = \{v_1, \dots, v_k\}$  ; L'élément  $\mathcal{C} = \{f_1, \dots, f_m\}$  est un ensemble de contraintes souples, où chaque  $f_i$  définit un coût dans  $\mathbb{R}^+ \cup \{+\infty\}$  pour chaque combinaison d'affectations à un sous-ensemble de variables (une contrainte est initialement connue des seuls agents concernés) ;  $\mu : \mathcal{X} \rightarrow \mathcal{A}$  fait correspondre les variables à leur agent ;  $f : \prod \mathcal{D}_{x_i} \rightarrow \mathbb{R}$  est une fonction objectif représentant le coût global d'une affectation complète de variables. L'objectif d'optimisation est représenté par la fonction  $f$ , qui, en général, est considérée comme la somme des coûts :  $f = \sum_i f_i$ . Une solution à un DCOP  $P$  est une affectation complète de toutes les variables. Une solution est *optimale* si elle minimise  $f$ .

Les DCOP ont été largement étudiés et appliqués à de nombreux domaines [8], car : (i) ils se concentrent sur des approches décentralisées dans lesquelles les agents coordonnent vers une solution commune par le biais d'échanges de messages locaux ; (ii) ils exploitent la structure du domaine (en l'encodant dans des contraintes) pour résoudre des problèmes de calcul difficiles ; (iii) il existe une grande variété de méthodes de résolution allant des méthodes exactes aux techniques heuristiques et

approximatives. Les méthodes de résolution utilisent des envois directs de messages qui, dans le contexte de notre UTM, sont effectués via un DMS.

### 4.2 Résoudre des conflits avec un DCOP

Modélisons maintenant notre problème de coordination en un DCOP qui doit être instancié chaque fois que certains UAV sont conscients d'un conflit. L'idée est double : (i) permettre aux UAV de choisir entre plusieurs actions de déconflition lorsqu'un conflit est détecté ; et (ii) faire en sorte que les UAV impliqués dans le même conflit se coordonnent pour choisir celui qui est le plus apte à effectuer la déconflition.

L'ensemble d'agents  $\mathcal{A} \subseteq U$  est l'ensemble des drones qui ont été alertés par le *service de détection des conflits* car impliqués dans certains conflits. Chacun de ces drones  $u$  connaît ses conflits  $C(u)$  et les autres drones avec lesquels il est en conflit,  $U(C(u)) = \{v \in U \mid v \neq u, C(u) \cap C(v) \neq \emptyset\}$ . Pour un conflit donné  $c$ , un drone est capable d'effectuer certaines actions de déconflition ; par exemple *postpone*( $c$ , 20), *elevate*( $c$ , -15), *elevate*( $c$ , +15) et *skip*( $c$ ). On note  $\mathcal{I}$  cet ensemble d'actions.

Les décisions consistent à choisir l'action de déconflition à déclencher pour chaque conflit connu. Notons  $x_{u,c,i} \in \{0, 1\}$  la variable de décision indiquant si le drone  $u$  décide de résoudre le conflit  $c$  en utilisant l'action  $i$ . Ainsi,  $\mathcal{X} = \{x_{u,c,i} \mid u \in \mathcal{A}, c \in C(u), i \in \mathcal{I}\}$ ,  $\mathcal{D} = \{d_{x_{u,c,i}} = \{0, 1\} \mid u \in \mathcal{A}, c \in C(u), i \in \mathcal{I}\}$ , et  $\mu : x_{u,c,i} \mapsto u$ .

Les contraintes se divisent en deux catégories : les coûts unaires (préférences pour les actions) et les coûts de coordination qui interdisent de résoudre le même conflit plus d'une fois. En ce qui concerne les coûts unaires, nous modélisons les préférences pour les actions générant moins de conflits sans diminuer la qualité du service (c-à-d. augmentant le nombre de points de passage manqués) et sans retarder la mission. Les agents doivent être capables d'évaluer les actions en utilisant le *service d'évaluation de la trajectoire* (TAS). Nous notons  $v_{\text{conf}}(i, c)$  (resp.  $v_{\text{miss}}(i, c)$ ) et  $v_{\text{delay}}(i, c)$  le nombre de conflits (resp. le nombre de points manqués et le retard) de la trajectoire lors de l'exécution de l'action  $i$  pour résoudre le conflit  $c$ . Nous agrégeons ces multiples évaluations objectives locales par linéarisation et hiérarchisation, comme suit, pour tous les  $u \in U$ ,  $c \in C(u)$  et  $i \in \mathcal{I}$  :

$$f_{\text{pref}}(x_{u,c,i}) = \omega^2 \cdot v_{\text{conf}}(i, c) + \omega \cdot v_{\text{miss}}(i, c) + v_{\text{delay}}(i, c) \quad (8)$$

avec  $\omega \in \mathbb{R}$  un nombre suffisamment grand pour donner la préférence à la résolution des conflits par rapport aux points de passage manqués et à la réduction des retards. Ces évaluations agrégées modélisent les objectifs multiples discutés dans la section 3. Concernant les contraintes de coordination, nous devons nous assurer que exactement une variable  $x_{i,c,u}$  est fixée à 1 pour un même conflit. Ainsi, pour chacun de ces conflits  $c$  :

$$\sum_{u \in U} \sum_{i \in \mathcal{I}} x_{i,c,u} = 1, \quad \forall c \in C(u) \quad (9)$$

Afin de s'intégrer dans le cadre de la DCOP, cette contrainte dure est codée en une contrainte souple comme suit :

$$f_{\text{exo}}(x_{u_1,c,i_1}, \dots, x_{u_{|\mathcal{I}|},c,i_{|\mathcal{I}|}}) = \begin{cases} 0, & \text{si (9)} \\ +\infty, & \text{sinon} \end{cases} \quad (10)$$

Par conséquent,  $\mathcal{C}$  est l'ensemble des contraintes de (8) et (10). La fonction objectif est la somme de toutes ces contraintes. Ce DCOP peut être résolu de manière distribuée par les agents concernés, en utilisant n'importe quelle méthode de résolution de la littérature, chaque fois que les agents doivent se coordonner.



FIGURE 3 – Un exemple de scénario sur une zone urbaine de 2 km x 2 km. 20 UAV sont représentés par des cercles (avec leur vecteur de vitesse). Les contrats 4D (et leurs tubes) sont représentés en vert. Les segments actuels sont représentés en bleu.

Dans le cas où un agent détecte un conflit avec une trajectoire non négociable (par exemple, une urgence), il essaiera toujours de résoudre le même problème mono-agent, en décidant quelle action est la meilleure.

## 5 Evaluation expérimentale

Nous évaluons maintenant la performance des comportements coordonnés et adaptatifs sur des scénarios denses, où plusieurs drones adaptent leurs trajectoires suite à l'occurrence d'incidents et l'ajout de trajectoires d'urgence.

### 5.1 Cadre expérimental

Nous analyserons les performances des comportements de déconfliction étudiés avec un nombre croissant d'UAV (et de trajectoires). L'environnement de simulation est codé en Java et exécuté sur un processeur 20-core Intel(R) Xeon(R) CPU E5-2660 v3 @ 2.60GHz, 62GB RAM, Ubuntu 18.04.5 LTS, avec une JVM OpenJDK 11.0.9.1. Nous avons exécuté 30 instances d'ensembles de trajectoires et d'incidents générés aléatoirement pour chaque taille de flotte, et nous traçons les valeurs moyennes, avec un intervalle de confiance de  $[0,05; 0,95]$ . Les algorithmes DCOP utilisés ici sont issus de la bibliothèque FRODO [12]. Les valeurs générées aléatoirement sont choisies uniformément dans les intervalles fournis. Le temps de calcul indiqué est le temps de simulation d'un processeur mono-CPU à 20 cœurs.

Nous considérons une zone de 2km par 2km, illustrée dans la figure 3, avec des plans verticaux d'espace aérien à 20m, 40m et 60m. Nous considérons des drones avec les caractéristiques suivantes :  $h_{max} = 18m.s^{-1}$ ,  $v_{max} = 6m.s^{-1}$ ,  $a_{max} = \Pi/2rad.s^{-1}$ ,  $\Delta h_{max} = \Delta v_{max} = 6m.s^{-2}$ ,  $\Delta a_{max} = \Pi/2rad.s^{-2}$ . La vitesse initiale est fixée à  $(0,0,0)$ . Les trajectoires initiales d'un UAV sont générées de manière aléatoire et incrémentale comme suit. Le point de départ est positionné aléatoirement dans l'espace aérien, à l'altitude  $z = 0$  et au temps  $t = 0$ . Les points suivants sont choisis aléatoirement dans une distance comprise entre 200m et 1000m, de manière à ce qu'il n'y ait pas de conflit avec les seg-

ments existants, en augmentant ou diminuant éventuellement l'altitude. Le temps pour le prochain point est également fixé en utilisant une vitesse de croisière par défaut, égale à  $(12,3,0)$  (équivalent à 40km/h horizontalement). Les UAVs ont initialement 60 points de passage vers leur destination, puis un dernier segment de retour vers leur origine, ce qui conduit à un ensemble dense de trajectoires et de croisements. Les tubes de sécurité sont définis par  $(h,v,t) = (30,15,1)$ , ce qui signifie que les UAV doivent être séparés d'au moins 30m horizontalement, 15m verticalement, avec 1s de tolérance temporelle. Pour ajouter des événements imprévisibles, nous générons 3 trajectoires d'urgence, consistant à partir d'une position aléatoire à la frontière de la zone, à se déplacer vers un point aléatoire à au moins 1km de distance du point initial, puis un point d'attente pendant 360s (pour simuler une extraction d'urgence), et enfin un retour au point initial. Ces trajectoires d'urgence ne sont pas modifiables. A chaque seconde simulée, il y a également une chance sur 1000 qu'un incident se produise à proximité d'un UAV choisi aléatoirement, devant ainsi capturer la scène pendant une durée aléatoire comprise entre 30s et 120s. Nous évaluons les comportements suivants : (i) postpone caractérise les drones qui n'effectuent qu'une action postpone( $c,20$ ) lorsqu'un conflit est identifié (abonnement aux services CDS et TUS); (ii) elevate caractérise les drones qui n'effectuent que des actions elevate( $c,\pm 15$ ), selon leur plan actuel, lorsqu'un conflit est identifié (abonnement aux services CDS et TUS); (iii) skip représente les drones qui n'exécutent l'action skip( $c$ ) que lorsqu'un conflit est détecté (abonnement aux services CDS et TUS); (iv) afb [9], dpop [18], dsa (variante C) [23], mgm2 [14] représentent des drones qui prennent une décision coordonnée et adaptative à l'aide d'un algorithme DCOP<sup>2</sup>, qui optent entre postpone( $c,d$ ) avec  $d \in \{20,40,60\}$ , elevate( $c,\pm 15$ ) et skip( $c$ ) (abonnement aux services DMS, CDS, TAS et TUS); et enfin, (v) centralized est un algorithme de déconfliction basé sur une recherche arborescente qui calcule la séquence optimale d'actions de réparation (du même ensemble que les DCOP) pour réparer complètement un ensemble de trajectoires conflictuelles, en optimisant l'objectif dérivé de l'équation (8). Tous les drones sont équipés d'une routine et d'un équipement de prévention des collisions. Le service CDS est configuré avec un horizon  $k = 10$ , ce qui dans notre cas représente environ 10min dans le futur.

La figure 4 présente les performances obtenues par simulation, avec un nombre croissant de drones; et la figure 5 montre l'évolution de certaines métriques, avec le temps, pour une instance spécifique avec 25 drones, 3 procédures d'urgence et 25 incidents.

### 5.2 Conflits et contrats

La première ligne de la figure 4 (4a-4d) porte sur les conflits et les trajectoires. Alors qu'en gestion du trafic aérien classique, avec un espace structuré, le report au départ est utilisé pour réduire les conflits [2], dans notre scénario à route libre, postpone génère de nombreux conflits en raison des retards accumulés sur les croisements de trajectoires, qui ne sont pas structurés comme des réseaux de flux, comme le montre la figure 4a. Tous les autres comportements génèrent encore des conflits (environ 200 sur les grandes instances), puisque près de 70 trajectoires sont ajoutées sur les grandes instances.

Face aux conflits, les UAV mettent à jour leurs contrats, qui ne sont plus garantis sans conflit. Ainsi, les drones ont tendance à demander de nombreuses mises à jour de contrats, comme l'illustre la figure 4b. C'est particulièrement vrai pour postpone et elevate, qui retardent les trajectoires d'une certaine durée,

2. Nous n'incluons pas l'algorithme MaxSum [6], car il ne peut pas converger dans la plupart de nos instances, fournissant ainsi des résultats de faible qualité tout en nécessitant de nombreux messages.

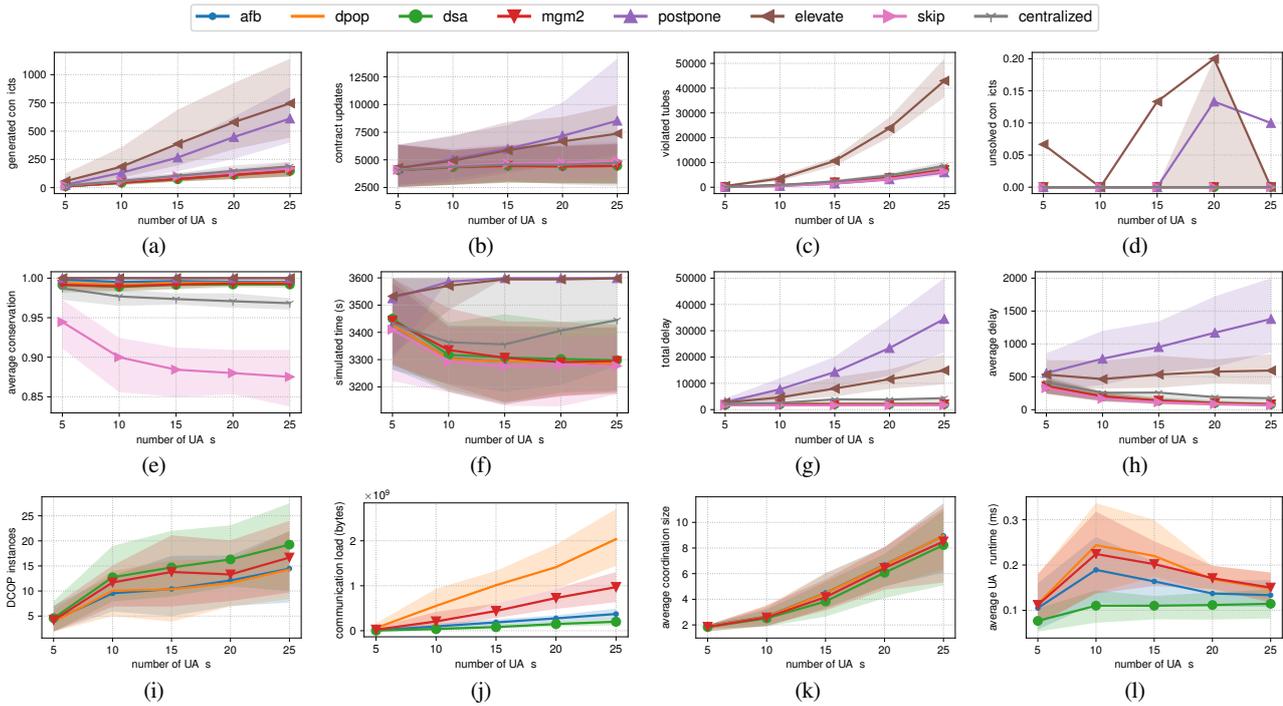


FIGURE 4 – Valeurs moyennes pour plusieurs mesures de performance avec un nombre croissant de drones

ce qui, la plupart du temps, n’est pas suffisant pour résoudre les conflits, ou qui génère d’autres conflits. Dans les grandes instances, une mise à jour de la trajectoire est demandée toutes les secondes, en moyenne. Cette dynamique de mise à jour des trajectoires fait que certains drones violent les tubes de sécurité (figure 4c), et déclenchent les routines anti-collision (figure 4d). En effet, lorsqu’un UAV demande la mise à jour d’un contrat avec un nouveau segment très proche de sa position actuelle (dans l’espace et le temps) et de la position d’un autre UAV, cela peut conduire à de telles situations, avant qu’un des deux UAV ne soit en mesure de mettre à jour à nouveau son contrat. Notamment, elevate fait que les UAVs violent 4 fois plus de tubes de sécurité que les autres comportements. Ces nombres de mises à jour et de violations sont élevés car nous choisissons ici de générer des trajectoires très denses, ce qui représente une sorte de pire scénario, où même les marges de sécurité ne sont pas suffisantes pour éviter les collisions.

En observant notre instance (figure 5), nous pouvons voir à quel point cette configuration est dynamique, ne laissant aucun répit aux drones pour réparer les trajectoires et remplir leurs missions, qui durent environ 1h. Il n’y a qu’une fenêtre "calme" entre  $t = 291$  et  $t = 744$ . Nous pouvons observer comment l’ajout d’incidents déclenche immédiatement des actions de report (que tous les agents mettent en œuvre pour gérer les incidents). En particulier, postpone et elevate génèrent de nombreux conflits juste après l’ajout d’une première trajectoire d’urgence, comme le montre la figure 5a. Face à ces conflits, les drones tentent constamment de réparer les trajectoires, jusqu’à ce qu’il n’y ait plus d’incidents et d’urgence dans l’espace aérien, comme l’illustre la figure 5b, après  $t = 2290$ . Même lorsque les contrats sont stables, les drones peuvent parfois violer certains tubes de sécurité, comme le montre la figure 5c, car ils ont été calculés à l’aide d’une projection temporelle peu précise.

### 5.3 Conservations des contrats et délais

La deuxième ligne de la figure 4 présente les métriques liées aux deuxième et troisième critères d’optimisation que nous

examinons, à savoir la conservation des trajectoires et les retards. Dans la figure 4e, qui indique le pourcentage moyen de points de passage initiaux conservés dans les trajectoires finales, skip est le comportement le moins conservateur (jusqu’à 12,5% de points sont abandonnés), car il contourne automatiquement les points de passage en cas de conflit, ce qui permet également de réduire le temps nécessaire pour terminer toutes les trajectoires (indiqué dans la figure 4f). Les deux comportements elevate et postpone ajoutent du retard pour résoudre les conflits (soit en changeant d’altitude, donc en passant du temps en altitude, soit en reportant les prochains points de passage), comme le montrent les figures 4g et 4h montrant le retard total accumulé et le retard moyen par UAV. Notons que chaque comportement ajoute du retard : cela est dû au fait que les drones traitent de nombreux incidents (jusqu’à 35 dans les plus grandes instances) les obligeant à rester en position pendant un certain temps, accumulant ainsi du retard. Le solveur centralisé perd en performance sur les grandes instances car il cherche d’abord à minimiser les conflits et les points de passage manqués, et reporte donc plus fréquemment que les solveurs basés sur DCOP. Les approches basées sur le DCOP co-optimisent localement les conflits, la conservation de la trajectoire et les retards, ce qui en fait un compromis efficace. En examinant notre instance isolée, nous pouvons observer comment skip saute constamment des points de passage pour répondre aux conflits, comme le montrent les figures 5d et 5e. skip supprime 3 fois plus de points que les comportements basés sur les DCOP. Les adaptations aux incidents ajoutent continuellement des retards aux trajectoires, en raison de l’action de report déclenchée dans de tels cas.

### 5.4 Mise en œuvre des DCOP

La troisième ligne de la figure 4 présente les métriques de fonctionnement des DCOP. Alors que les trajectoires générées sont denses dans une zone limitée, les procédures DCOP entre plusieurs drones sont assez peu fréquemment déclenchées, comme le montre la figure 4i. Cela s’explique par le fait que les décisions coordonnées basées sur la DCOP parviennent à

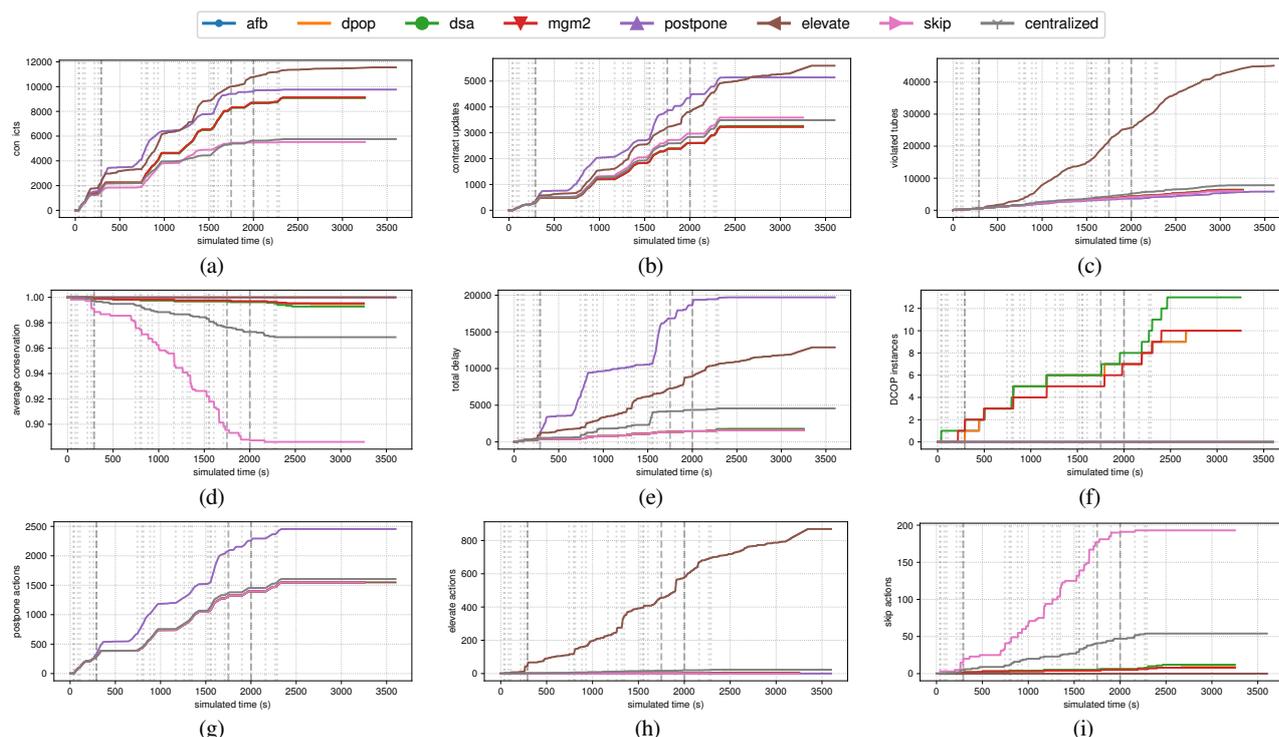


FIGURE 5 – Évolution des mesures de performance au cours d'une simulation avec 25 drones, 3 procédures d'urgence (lignes pointillées grises) et 25 incidents (lignes pointillées grises)

résoudre les conflits pour l'horizon donné, puis ne nécessitent pas d'autre coordination pendant un certain temps, sauf si d'autres perturbations se produisent. Il est intéressant de noter que les solveurs incomplets (*dsa*, *mgm2*) nécessitent plus de coordination que les solveurs complets (*afb* et *dpop*). En effet, comme les décisions prises par ces solveurs incomplets ne sont pas optimales, elles peuvent donner lieu à des décisions conflictuelles nécessitant une coordination supplémentaire de déconfliction. Pendant la coordination, les conflits concernent en moyenne un tiers de la flotte totale, comme le montre la figure 4k. La coordination peut théoriquement regrouper de nombreux agents, mais dans notre cadre, la taille restreinte des DCOP permet d'utiliser des méthodes de solution complètes comme *afb* et *dpop*, ce qui n'a pas d'impact important sur le temps d'exécution de la décision individuelle du drone, comme le montre la figure 4l. Si l'on considère la charge de communication dans la figure 4j, *dpop* nécessite l'échange de messages peu nombreux mais de grande taille puisque les DCOP à résoudre sont cycliques et nécessitent donc l'échange de messages de taille exponentielle. *mgm2*, en raison de ses paramètres de coordination à 2, échange de nombreux petits messages.

Les comportements basés sur les DCOP sont déclenchés lorsqu'ils font face à certaines des perturbations, comme le montre la figure 5f, et appliquent les actions avec les meilleures évaluations, à savoir *elevate* (pour conserver les points de passage) et *skip* (pour limiter les retards), comme le montrent les figures 5h et 5i. L'action *postpone* n'est pas utilisée dans ces conditions, comme on peut l'observer dans la période "calme" de la figure 5g. Cela explique comment les DCOP obtiennent de bons résultats en tant que compromis entre la conservation du contrat et les retards. *afb* est le meilleur concurrent, avec une bonne déconfliction nécessitant une faible charge de communication. Cependant, *afb* n'est pas robuste à la perte de messages, contrairement à *dsa*. D'autres expérimentations

devraient être menées pour évaluer l'impact du manque de fiabilité des communications sur les performances du DCOP.

## 5.5 Résumé

Ces résultats soulignent que dans la configuration étudiée extrêmement dynamique, les drones doivent constamment mettre à jour leurs trajectoires. Il est clair que le comportement *postpone* est mauvais sur deux dimensions : les délais et les conflits (donc la sécurité). Le comportement *elevate* est un bon candidat pour la conservation de la trajectoire, mais avec des retards supplémentaires et de nombreuses violations de tubes, tandis que *skip* génère peu de conflits, au prix de quelques points de passage manqués. Le comportement basé sur un DCOP se positionne comme un bon compromis entre *elevate* et *skip*, répondant ainsi à notre objectif de minimiser les conflits générés, de maximiser la conservation de la trajectoire et de minimiser les retards – de la même qualité qu'une déconfliction centralisée. Dans d'autres contextes, avec de nombreux carrefours sur des espaces aériens structurés, ces résultats pourraient ne pas être valables. En effet, une grande partie des performances dépend du fait que les UAV peuvent librement changer d'altitude ou contourner certains points de passage. Dans un espace aérien plus structuré, les UAV changent normalement d'altitude dans certains couloirs verticaux limités, et ne peuvent pas voler en dehors des couloirs horizontaux, comme défini dans [4], par exemple. Dans de tels cas, *postpone* se comporterait probablement mieux, tandis que les DCOP formeraient de plus grands groupes de trajectoires conflictuelles, passant par un nombre limité de carrefours. Nous gardons cette investigation et cette analyse pour des travaux futurs.

## 6 Conclusion

Dans cet article, nous avons promu et mis en œuvre un mécanisme de coordination multi-agents pour aider les drones évoluant dans un espace aérien dynamique à route libre à adapter leurs

trajectoires. La coordination est construite sur un protocole distribué de résolution de contraintes, qui vise à minimiser les conflits futurs, à maximiser la conservation des trajectoires initiales, et à réduire le délai global impliqué par les actions de déconflition. Sur le scénario simulé que nous avons développé, formant un espace aérien dense avec des événements aléatoires à gérer, nous montrons que l'utilisation d'une déconflition coordonnée et adaptative, fournit des solutions avec moins de conflits, de points de passage manqués et de retard accumulé, par rapport aux solutions non coordonnées et fixes. Cependant, cette coordination nécessite l'échange de messages de communication supplémentaires. Dans l'ensemble, l'approche basée sur le DCOP est un candidat prometteur pour l'installation de la déconflition autonome dans l'UTM, et ouvre la voie à de futures recherches.

Puisque les visions de l'UTM et de l'U-Space évoluent, notamment vers des espaces aériens plus structurés [3, 4], nous envisageons d'étendre notre cadre et nos services à de tels contextes, qui promettent d'être plus difficiles à résoudre en raison du manque de liberté de trajectoire et de la plus grande taille des groupes de déconflition. Nous souhaitons en particulier adapter la méthode de résolution DCOP aux situations de déconflition de trajectoire. En fait, nous avons utilisé ici une simple agrégation linéaire de trois critères, qui génère toujours des trajectoires non conflictuelles. Les DCOPs multi-objectifs pourraient être une approche pertinente pour des développements futurs [15]. En outre, la conception d'algorithmes de coordination moins gourmands en communications semble être une condition nécessaire au déploiement de ces protocoles et services sur des dispositifs réels et des UTM. Cela pourrait se faire soit en limitant la taille des DCOP de déconflition, soit en s'appuyant sur des méthodes de solution légères et robustes comme A-DSA [20], au détriment de l'optimalité.

## Références

- [1] D. Alejo, J. A. Cobano, G. Heredia, and A. Ollero. Particle Swarm Optimization for collision-free 4D trajectory planning in Unmanned Aerial Vehicles. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 298–307, 2013.
- [2] N. Barnier and C. Allignol. Trajectory deconflition with constraint programming. *The Knowledge Engineering Review*, 27(3) :291–307, 2012.
- [3] A. Bauranov and J. Rakas. Designing airspace for urban air mobility : A review of concepts and approaches. *Progress in Aerospace Sciences*, 125 :100726, 2021.
- [4] C. Capitán, H. Pérez-León, J. Capitán, A. Castaño, and A. Ollero. Unmanned Aerial Traffic Management System Architecture for U-Space In-Flight Services. *Applied Sciences*, 11(9), 2021.
- [5] D. Delahaye, S. Puechmorel, P. Tsiotras, and E. Feron. Mathematical Models for Aircraft Trajectory Design : A Survey. In *Air Traffic Management and Systems*, pages 205–247, Tokyo, 2014. Springer Japan.
- [6] A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings. Decentralised Coordination of Low-Power Embedded Devices Using the Max-Sum Algorithm. In *AAMAS 2008*, pages 639–646, 2008.
- [7] Federal Aviation Agency. UTM concept of operations version 2.0, 2020.
- [8] F. Fioretto, E. Pontelli, and W. Yeoh. Distributed Constraint Optimization Problems and Applications : A Survey. *Journal of Artificial Intelligence Research*, 61 :623–698, 2018.
- [9] A. Gershman, A. Meisels, and R. Zivan. Asynchronous Forward-Bounding for Distributed Constraints Optimization. In *ECAI 2006*, page 103–107, NLD, 2006. IOS.
- [10] Y. Hamadi. Optimization for Urban Air Mobility. In I. S. Kotsireas and P. M. Pardalos, editors, *Learning and Intelligent Optimization*, pages 1–8, Cham, 2020. Springer International Publishing.
- [11] A. Islami, S. Chaimatanan, and D. Delahaye. *Large-Scale 4D Trajectory Planning*, pages 27–47. Springer Japan, Tokyo, 2017.
- [12] T. Léauté, B. Ottens, and R. 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>.
- [13] 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.
- [14] R. T. Maheswaran, J. P. Pearce, and M. Tambe. Distributed Algorithms for DCOP : A Graphical-Game-Based Approach. In *ISCA PDCS*, pages 432–439, 2004.
- [15] T. Matsui, M. Silaghi, K. Hirayama, M. Yokoo, and H. Matsuo. Distributed Search Method with Bounded Cost Vectors on Multiple Objective DCOPs. In I. Rahwan, W. Wobcke, S. Sen, and T. Sugawara, editors, *PRIMA 2012 : Principles and Practice of Multi-Agent Systems*, pages 137–152. Springer, 2012.
- [16] C. Nava-Gaxiola, C. Barrado, and P. Royo. Study of a Full Implementation of Free Route in the European Airspace\*. In *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, pages 1–6, 2018.
- [17] G. Perozzi, D. Efimov, J.-M. Biannic, and L. 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.
- [18] A. Petcu and B. Faltings. A scalable method for multiagent constraint optimization. In *International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 266–271, 2005.
- [19] G. Picard. Trajectory Coordination based on Distributed Constraint Optimization Techniques in Unmanned Air Traffic Management. In *AAMAS 2022*, pages 1065–1073, 2022.
- [20] P. Rust, G. Picard, and F. Ramparany. Resilient Distributed Constraint Optimization in Physical Multi-Agent Systems. In *European Conference on Artificial Intelligence (ECAI)*. IOS Press, 2020.
- [21] SESAR. U-Space : Concept of Operations. Technical report, SESAR Joint Undertaking, 2019.
- [22] S. Yan and K. Cai. A multi-objective multi-memetic algorithm for network-wide conflict-free 4D flight trajectories planning. *Chinese Journal of Aeronautics*, 30(3) :1161–1173, 2017.
- [23] W. Zhang, G. Wang, Z. Xing, and L. Wittenburg. Distributed Stochastic Search and Distributed Breakout : Properties, Comparison and Applications to Constraint Optimization Problems in Sensor Networks. *Artificial Intelligence*, 161(1–2) :55–87, 2005.
- [24] W. Zhao, Z. Fang, and Z. Yang. Four-Dimensional Trajectory Generation for UAVs Based on Multi-Agent Q Learning. *Journal of Navigation*, 73(4) :874–891, 2020.