

# Algorithme évolutionnaire pour la résolution, en continu, de conflits aériens

Sarah Degaugue<sup>1,2</sup>, Nicolas Durand<sup>1</sup>, Jean-Baptiste Gotteland<sup>1</sup>

<sup>1</sup> École Nationale de l'Aviation Civile, France

{sarah.degaugue,nicolas.durand,jean-baptiste.gotteland}@recherche.enac.fr

<sup>2</sup> DSN-ATI/Domaine Innovation

**Mots-clés** : *recherche opérationnelle, optimisation, problèmes dynamiques, algorithme évolutionnaire.*

## 1 Introduction

### 1.1 Problématique

En raison des incertitudes dues aux conditions météorologiques, aux prédictions de trajectoire et à l'évolution du trafic aérien, contrôler un secteur aérien est un problème dynamique. L'objectif à long terme est d'offrir aux contrôleurs aériens un outil d'aide à la décision intelligent adapté au caractère dynamique du trafic. La résolution successive d'une situation de trafic au cours du temps n'a encore jamais été étudiée. Une comparaison préliminaire a toutefois été réalisée et des conclusions prometteuses quant à l'ajout d'une mémoire ont été faites ([1]). Cependant, l'algorithme évolutionnaire au cœur de la résolution ayant été amélioré depuis, les résultats et l'efficacité des approches ont évolué.

Dans cet article, nous étudions différentes utilisations d'un algorithme évolutionnaire qui a pour mission de détecter et de résoudre les conflits dans un secteur aérien en cherchant les paramètres de l'algorithme qui conviennent le mieux pour chaque approche.

### 1.2 État de l'art

Dans la recherche des Problèmes Dynamiques Évolutionnaires (PDE) actuelle où les problèmes considérés varient dans le temps, l'objectif d'un algorithme passe de la recherche de l'optimum global à la détection des changements, puis au suivi de l'évolution des optimums (optimums locaux ou idéalement l'optimum global) au fil du temps. De plus, dans le cas présent, où le problème après le changement est en corrélation avec le problème avant le changement, un algorithme d'optimisation doit également apprendre autant que possible de son expérience de recherche précédente pour, espérons-le, faire avancer la recherche plus efficacement.

L'approche la plus naïve pour résoudre les Problèmes Dynamiques d'Optimisation (PDO) est de réinitialiser le processus évolutif lorsqu'un changement se produit. Une approche similaire redémarre la population basée sur la convergence d'algorithmes évolutionnaires ([4]). Le défaut de ces approches est que presque aucun des matériaux évolutifs passés n'est jamais utilisé, alors que cela pourrait accélérer la convergence.

Les approches de mémoire explicite sont des approches qui utilisent la mémoire externe pour stocker des éléments antérieurs qui peuvent être utiles dans les étapes futures du processus évolutif. Ces approches utilisent couramment une mémoire de petite taille pour stocker les meilleures solutions et les ajouter à la population si elles sont mieux adaptées que les individus actuels ([6],[7],[8]). Trojanowski et Michalewicz comparent quatre différentes façons de procéder

en cas de changement dans l’environnement de recherche. La première consiste à faire évoluer l’algorithme évolutionnaire comme usuellement. La seconde consiste à associer à chaque chromosome une mémoire de taille constante : une fois un individu créé, sa mémoire va contenir son parent, ou le meilleur des deux s’il en a deux, et les ancêtres de ce dernier. Lorsqu’un changement survient dans l’environnement, chaque chromosome actif et ses ancêtres sont réévalués et si un ancêtre s’avère être meilleur que son chromosome actif *enfant*, il est sélectionné à sa place. La troisième méthode comparée consiste à ré-initialiser une certaine part des éléments de la population courante au moment du changement (environ 85% des éléments sont réinitialisés dans le document). Enfin, la dernière approche regroupe à la fois la mémoire explicite des ancêtres, et l’initialisation de certains individus. Les résultats montrent globalement que la combinaison d’une mémoire explicite et d’une ré-initialisation des individus est meilleure pour le cas d’étude en termes de *précision* (l’algorithme trouve une solution proche de l’optimum avant chaque changement d’environnement) et d’*adaptabilité* (l’algorithme trouve en moyenne une solution proche de l’optimum sur toutes les générations).

Dans cet article, nous étudions les conséquences de l’ajout d’une mémoire explicite au processus évolutionnaire dans le cadre d’une résolution de conflits aériens continue, en comparant le paramétrage optimal de l’algorithme pour chaque approche étudiée.

## 2 Modélisation

Habituellement, les plans de vol se résument à des successions de balises que doivent suivre les pilotes. Dans ces premières simulations, on considère des trajectoires directes d’un point d’origine O à un point de destination final D, ces points pouvant être apparentés à des balises.

### 2.1 Détection de conflit

Un conflit correspond à un risque de perte de séparation entre deux aéronefs. Étant donné que nous travaillons uniquement dans le plan horizontal dans cette analyse, une unique distance de 5 NM doit être respectée horizontalement.

Un modèle d’incertitude isotropique s’ajoute aux calculs en augmentant cette norme dans le temps de prédiction avec un taux de 5% de la somme des vitesses des deux avions. À partir d’un certain horizon  $t_h$ , la norme de séparation est bornée à  $d_t = d_h$  pour  $t \geq t_h$ .

### 2.2 Manœuvre

Au fur et à mesure d’une simulation, des manœuvres vont être appliquées aux avions. Dans le modèle de trajectoire choisi, les manœuvres associées aux différents avions d’un secteur sont des changements de cap d’ $\alpha$  degrés qui commencent au temps  $t_0$  et qui se terminent au temps  $t_1$  (voir Figure 1).

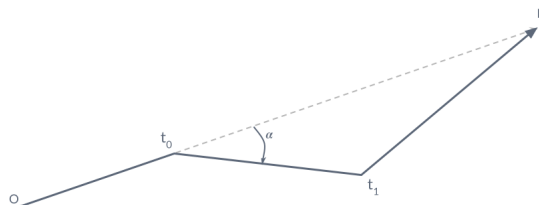


FIG. 1 – Changement de cap  $\alpha$  commençant au temps  $t_0$  et terminant au temps  $t_1$

À un instant donné, un avion peut être en cours de manœuvre, ce qui réduit les domaines de ses variables  $(t_0, \alpha, t_1)$  : une fois  $t_0$  dépassé, seul  $t_1$  pourra encore varier et une fois  $t_1$  passé, l’avion ne pourra plus être manœuvré.

## 3 Algorithme évolutionnaire (AE)

### 3.1 Principe général

L'algorithme évolutionnaire se rapproche des algorithmes génétiques classiques. La fin de notre AE est acté par une stabilisation du score du meilleur individu trouvé depuis 20 générations, ainsi que par un nombre maximal absolu de générations. Des opérations de mutations et de croisements pour diversifier et intensifier l'optimisation sont également utilisées. Elles exploitent notamment les *fitness partielles* (ou *fitness locales*, définies dans la partie 3.2) de chaque individu, tel que décrit dans [2], pour cibler les variations les plus prometteuses. Un processus de *sharing* est également mis en place pour pénaliser les individus trop proches.

### 3.2 Élément de population

Considérons une situation donnée : au temps courant  $t_c$ , un secteur contient  $n$  avions.

Un élément de population, aussi appelé *chromosome*, est un ensemble de  $n$  manœuvres associées aux différents avions de la simulation. Ces manœuvres pourront être appelées *allèles* par abus de langage, faisant référence au lexique biologique associé à un chromosome.

On associe à chaque individu un score, appelé *fitness*, que nous cherchons à maximiser, et permettant de comparer les différents éléments de population entre eux. Cette *fitness* est calculée à l'aide des *fitness locales* associées à chaque allèle d'un chromosome.

Le premier objectif est qu'il n'y ait aucun conflit entre les avions. Les expressions des fitness locales et de la fitness de l'individu sont respectivement (1) et (2), avec  $conf_i$  l'ensemble des conflits relatifs à l'avion  $i$  et  $duree_c$  la durée du conflit  $c \in conf_i$ .

$$(1) f_i = \frac{1}{1 + \sum_{c \in conf_i} duree_c} \quad (2) F = \frac{1}{n} \sum_{i=1}^n f_i$$

À l'inverse, si l'individu n'engendre plus aucun conflit, nous allons chercher à minimiser le nombre de manœuvres et les délais qu'engendrent les manœuvres choisies. En outre, les manœuvres doivent débuter le plus tard possible pour avoir l'opportunité de les annuler ou d'en exécuter d'autres si la situation vient à évoluer. Dans ce cas, les expressions des fitness locales et de la fitness de l'individu sont respectivement (3) et (4), avec  $delais_i$  la différence entre la durée de la nouvelle route et la durée du parcours direct du point d'origine au point de destination,  $t_{0,max_i} = t_{0,i}$  si l'avion n'est pas manœuvré,  $\lfloor f_i \rfloor$  la partie entière de la fitness  $f_i$  valant 0 si l'avion est manœuvré, 1 sinon, et enfin,  $\{f_i\}$  la partie fractionnaire de la fitness  $f_i$  prenant en considération le retard et le temps de départ de la manœuvre associée à l'avion  $i$ .

$$(3) f_i = \frac{1}{1 + 2 * delais_i + (t_{0,max_i} - t_{0,i})} \quad (4) F = 1 + \sum_{i=1}^n \lfloor f_i \rfloor + \frac{1}{n} \sum_{i=1}^n \{f_i\}$$

## 4 Différentes utilisations de l'algorithme évolutionnaire

### 4.1 Approche naïve (NA)

En premier lieu, nous choisissons une approche simple qui consiste à appeler un solveur de conflits périodiquement pendant la simulation. Toutes les 30 secondes, le simulateur de trafic appelle le solveur de conflits. Une fois que celui-ci a fini son exécution, le simulateur reçoit la meilleure solution trouvée.

### 4.2 Approche à mémoire explicite (AME)

L'idée de l'approche à mémoire explicite est de garder des informations trouvées lors des anciennes optimisations. Contrairement à l'approche naïve, l'AME essaie de construire de nouvelles solutions proches des anciennes. En effet, après chaque résolution effectuée par le solveur, la dernière génération est conservée pour initialiser la population de la prochaine optimisation.

Seuls les éléments toujours compatibles avec les nouvelles contraintes de l'environnement courant sont gardés. Trois cas sont possibles :

- un nouvel avion est arrivé dans le secteur : un nouvel allèle correspondant à la manœuvre de cet avion va être aléatoirement initialisé et ajouté à chaque élément de population ;
- un avion a terminé sa manœuvre ou a quitté le secteur : l'allèle correspondant à la manœuvre de cet avion est supprimé pour chaque élément de population ;
- un avion a commencé sa manœuvre depuis la dernière optimisation : l'allèle correspondant devient contraint ( $t_0$  et  $\alpha$  fixés).

### 4.3 Approche continue (AC)

L'objectif de cette approche est de faire vivre en continu une population d'individus et de l'adapter régulièrement aux changements de l'environnement courant. Ainsi, la fonction d'évaluation et les contraintes énoncées en partie 4.2 évoluent constamment pendant le déroulement de l'AE.

Le simulateur peut alors avoir accès, dès qu'il le souhaite, à la meilleure solution courante de l'algorithme évolutionnaire.

Pour tester cette approche, nous avons choisi d'autoriser 400 générations maximales lors de la première résolution (début de simulation) puis seulement 40 générations maximales toutes les 3 secondes pendant le reste de la simulation.

## 5 Paramétrage

Dans cette section, nous allons chercher à sélectionner, pour les trois approches, les paramètres ( $nb_e$ ,  $p_{cross}$ ,  $p_{mut}$ ) de l'algorithme les mieux adaptés, avec  $nb_e$  le nombre d'éléments de population par génération qui varie entre 50 et 200 par pas de 50,  $p_{cross}$  la part d'éléments de population croisés choisie entre 0,1 et 0,7 par pas de 0,1, et  $p_{mut}$  la part d'éléments de population mutés variant entre 0,1 et 0,7 par pas de 0,1. De plus, nous fixons le nombre de générations maximal à  $nb_g = 400$ .

Les trois approches sont évaluées sur 10 exercices de maximum 600 secondes, faisant intervenir entre 6 et 9 avions.

### 5.1 Analyse préliminaire

Dans un premier temps, nous retenons les paramètres permettant de résoudre tous les conflits sur toutes les résolutions. Parmi les 112 jeux de paramètres parcourus, l'approche à mémoire explicite EMA en conserve 77, l'approche continue 74 et enfin l'approche naïve 65. L'approche avec mémoire explicite se révèle ainsi la plus robuste aux variations de paramètres de l'AE.

Pour la suite de l'analyse, seulement ces jeux de paramètres seront gardés.

### 5.2 Observation globale

Pour comparer l'efficacité des jeux de paramètres, nous allons maintenant nous intéresser aux fitness moyennes **FM** obtenues sur chaque pas de temps et pour tous les exercices, et aux produits des nombres de générations *utiles* moyennes par le nombre d'éléments par population **GUMxEP**. Le nombre de générations *utiles* est le nombre de générations à partir duquel la fitness ne s'est plus améliorée.

Sur la Figure 2 chaque point correspond à un jeu de paramètres placé en abscisse par le GUMxEP, et en ordonnée par la FM, et sur la Figure 3 chaque point correspond à un jeu de paramètres placé en abscisse par le produit GUMxEP, et en ordonnée par la PAM.

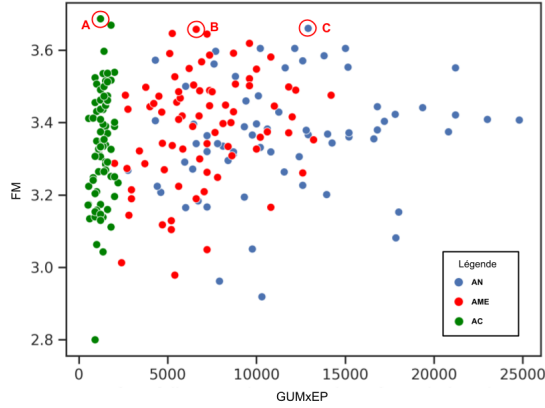


FIG. 2 – Fitness moyenne

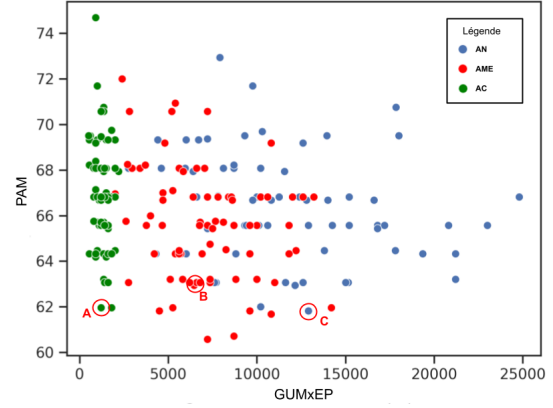


FIG. 3 – Pourcentage d'avions manœuvrés

Sur les deux figures, les points verts représentent les jeux de paramètres conservés pour l'approche continue, les points rouges ceux de l'approche à mémoire explicite et les points bleus ceux de l'approche naïve.

Nous constatons rapidement que les résolutions à l'aide de l'AC sont les moins coûteuses en nombre d'opérations, pour des solutions de bonnes qualités. Après la première résolution, 40 générations suffisent toujours à résoudre toutes les situations. L'approche naïve coûte globalement plus *cher* que les deux autres versions.

Intéressons-nous maintenant aux points A, B, C mis en évidence sur la Figure 2. Ces points représentent les meilleurs jeux de paramètres en termes de FM pour chaque version (A pour AN, B pour AME, C pour AC). À un instant  $t$ , si l'algorithme propose  $n_m$  manœuvres supplémentaires, le simulateur ne va pas obligatoirement toutes les appliquer : seules les manœuvres commençant dans moins de 60 secondes sont appliquées aux avions concernés. Ainsi, la *fitness* associée à cette résolution n'est pas forcément entièrement indicatrice de la part d'avions manœuvrés au cours de la simulation. C'est pourquoi nous allons regarder de plus près le pourcentage exact d'avions manœuvrés moyen **PAM**, présent en Figure 3.

Les points A, B et C identifiés précédemment se retrouvent globalement dans les meilleurs jeux de paramètres lorsque l'on s'intéresse au pourcentage réel moyen d'avions manœuvrés. Étant donné que ce critère est prioritaire dans l'expression de la *fitness* lorsqu'il n'y a plus de conflit, ce résultat est plutôt cohérent.

## 6 Validation des paramètres retenus

À partir des résultats précédents, nous avons choisi pour chaque approche moins de 10 jeux de paramètres à tester sur 10 scénarios longs (environ 12000 secondes) à flux de trafic non constant. Les effets à long terme sont alors plus conséquents que pour les scénarios précédents, mais cette étape d'observation est primordiale pour valider ces approches car elles visent à être employées pour des situations de contrôle continues à flux de trafic variable.

Alors que les différentes approches (NA et AME) ne résolvaient pas certains conflits lors de l'étude préliminaire ([1]), cette fois, et pour chaque approche, un jeu de paramètres permet de résoudre tous les conflits dans tous les exercices longs.

Ces jeux de paramètres et leurs résultats associés ont été indiqués dans le Tableau 1.

La différence particulièrement notable entre ces 3 approches est le coût de calcul des solutions, illustré par le produit GUMxEP, qui est particulièrement faible pour la version continue comme indiqué dans la partie précédente.

Approche	$nb_e$	$p_{cross}$	$p_{mut}$	FM	PAM	GUMxEP
NA	200	0.4	0.1	4.70	58.3	9800
AME	100	0.7	0.2	4.88	53.9	4900
AC	200	0.5	0.4	4.81	55.7	2800

TAB. 1 – Meilleurs jeux de paramètres résolvant tous les conflits

## 7 Conclusions et perspectives

Dans l’optique de proposer aux contrôleurs aériens un outil de résolution de conflits, nous devons mettre en place un solveur capable de résoudre rapidement et efficacement une situation donnée. Les évolutions effectuées sur l’algorithme évolutionnaire ont permis de mettre en œuvre une approche naïve plutôt efficace en termes de pourcentage d’avions manœuvrés. Cependant, les approches avec mémoire permettent de réduire davantage ce pourcentage, notamment à l’aide de la mémoire explicite où ce pourcentage est le plus faible et la fitness moyenne est la plus forte. L’approche continue a l’avantage de donner des solutions correctes en très peu de temps pendant l’évolution des situations de trafic.

Nous visons désormais l’intégration de ces versions dans un simulateur réaliste de trafic aérien incluant la notion de secteur aérien et la dimension verticale, pour avoir des retours de professionnels du contrôle aérien et ainsi valider et sélectionner l’approche qui leur convient le mieux.

## Références

- [1] Sarah Degaugue, Nicolas Durand et Jean-Baptiste Gotteland. Impact of Explicit Memory on Dynamic Conflict Resolution. *ICRAT*, 2022.
- [2] Nicolas Durand et Jean-Marc Alliot. Genetic operators adapted to partially separable functions. *Technical Report CENA*, 1996.
- [3] David E. Goldberg et Robert Elliott Smith. Nonstationary Function Optimization Using Genetic Algorithms with Dominance and Diploidy. *ICGA*, 1987.
- [4] Kalmanje Krishnakumar. Micro-Genetic Algorithms For Stationary And Non-Stationary Function Optimization. *Proc. SPIE 1196, Intelligent Control and Adaptive Systems* :289–296, 1990.
- [5] Khim Peow Ng et Kok Cheong Wong. A New Diploid Scheme and Dominance Change Mechanism for Non-Stationary Function Optimization. *In Proceedings of the 6th International Conference on Genetic Algorithms. Morgan Kaufmann Publishers Inc.* :159–166, 1995.
- [6] Sushil Louis et Zhijie Xu. Genetic Algorithms for Open Shop Scheduling and Re-Scheduling. *ISCA 11th Intl. Conf. on Computers and their Applications* :99–102, 1996.
- [7] K. Trojanowski et Z. Michalewicz. Searching for optima in non-stationary environments. *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99*, Vol. 3 :1843–1850, 1999.
- [8] K. Trojanowski, Z. Michalewicz et Jing Xiao. Adding memory to the Evolutionary Planner/Navigator. *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC ’97)* :483–487, 1997.