

Un algorithme de colonie de fourmis pour résoudre des conflits aériens

Nicolas Durand^{1,2}, Jean-Marc Alliot^{1,2}

¹ DSNA/DTI R&D[‡], 7, av Edouard Belin, BP 54005 F-31055 Toulouse Cedex 4, France
durand,alliot@tls.cena.fr

² APO-IRIT[§], Université Paul Sabatier ; 118, route de Narbonne, F-31062 Toulouse Cedex 9, France

Mots-Clés : algorithme colonie de fourmis, métaheuristique, résolution de conflits

1 Introduction

Le problème de résolution de conflits aériens en croisière est un problème très combinatoire impossible à résoudre avec des algorithmes d'optimisation classiques dans un contexte réaliste : d'une part l'évaluation d'une solution ne peut se faire que par une simulation prenant en compte des modèles de prévision de trajectoires complexes ; d'autre part, les variables en jeu sont en général discrètes et leur nombre peut être très élevé. La littérature ne propose que deux approches efficaces pour résoudre de façon centralisée des problèmes de grande taille (plus d'une vingtaine d'avions). L'approche de [PFB02] utilise la programmation linéaire mixte, mais requiert des hypothèses fortes sur les trajectoires (vitesses constantes, manoeuvres exécutées en même temps). L'approche par algorithme évolutionnaire [DA98] est plus ancienne et le présent article propose, toujours dans le domaine des métaheuristicques, une alternative à cet d'algorithme. Cet article a pour but de présenter un algorithme de colonie de fourmis, tel que décrit initialement dans [DMC96], adapté à un problème de résolution de conflits de grande taille. Un problème "jouet" a été utilisé pour la mise au point de l'algorithme. n avions sont disposés sur un cercle et se dirigent vers le centre du cercle (figure 1).

[‡]Direction des Services de la Navigation Aérienne, Direction de la Technique et de l'Innovation, Domaine R & D

[§]Equipe Algorithmes Parallèles et Optimisation - Institut de recherche en Informatique de Toulouse

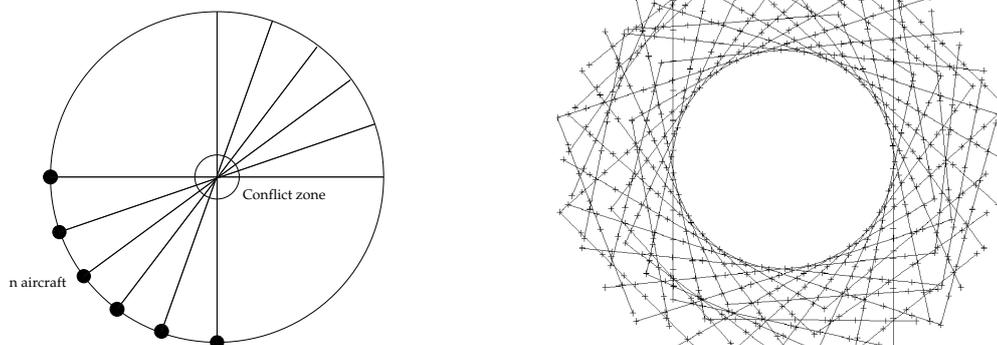


Fig. 1 – n avions convergent vers le centre du cercle. A droite : solution optimale pour 30 avions.

2 Modélisation du problème

Le temps est discrétisé par pas d'une minute et les avions peuvent tourner à droite ou à gauche de 10, 20 ou 30 degrés à chaque pas de temps. Quand l'avion a changé une fois de cap, il n'a que la possibilité revenir vers sa destination. Si l'on dispose de n_t pas de temps, on calcule que chaque avion a $12 \times n_t - 5$ trajectoires possibles. L'espace des solutions est ainsi composé de $(12 \times n_t - 5)^n$ combinaisons de trajectoires qui nécessitent la mise à jour d'autant de pistes de phéromones si l'on modélise une solution par une fourmi. L'idée originale de ce papier est de modéliser chaque avion par une fourmi ce qui réduit le nombre de pistes de phéromones à mettre à jour à $n(12 \times n_t - 5)$.

3 Description de l'algorithme

La quantité initiale de phéromones est répartie uniformément sur tous les chemins possibles. Un coût est attribué à chaque chemin. Le coût est nul tant qu'il n'y a pas de déviation, en phase de changement de cap, il est de +2 par pas de temps et il est ramené à +1 dans la phase de retour vers la destination. A chaque itération, un groupe de fourmis représente chaque avion. Chaque fourmi qui atteint sa destination sans rencontrer de conflit déposera une quantité de phéromones supplémentaire sur son chemin inversement proportionnelle à la longueur de ce dernier. Une évaporation est effectuée sur chaque piste de phéromones après chaque itération. Dans des configurations très complexes, il se peut qu'aucune fourmi n'atteigne sa destination sans rencontrer de conflit. Dans ce cas, l'algorithme ne peut progresser vers une solution. L'algorithme propose alors de relaxer la contrainte de conflits pour permettre aux fourmis qui ont rencontré le moins de conflits de déposer des phéromones sur les chemins les plus prometteurs. Lorsqu'un nombre important de fourmis arrive à respecter la contrainte relaxée, cette dernière est renforcée.

4 Résultats

Un exemple de conflit à 30 avions est résolu sur la figure 1. Avec une centaine de générations, 30 fourmis par avion, on trouve la solution optimale du problème en quelques secondes. Plusieurs paramètres doivent être ajustés, tels que le taux d'évaporation, le nombre de fourmis par avion, la stratégie de relaxation de la contrainte. La suite de ce travail exploratoire consistera à tester l'algorithme sur une échelle plus grande, avec un simulateur de trafic aérien utilisant des données de trafic réel, afin de vérifier la capacité de résolution de l'algorithme sur un grand nombre d'exemples et de comparer ses performances avec celles obtenues avec un algorithme évolutionnaire.

Références

- [DA98] Nicolas Durand and Jean-Marc Alliot. Genetic crossover operator for partially separable functions. In *Genetic Programming, 1998*.
- [DMC96] Marco Dorigo, Vittorio Maniezzo, and Alberto Coloni. The ant system : Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26 :29–41, 1996.
- [PFB02] L. Pallottino, E. Feron, and A. Bicchi. Conflict resolution problems for air traffic management systems solved with mixed integer programming. *IEEE Transactions on Intelligent Transportation Systems*, 3(1) :3–11, 2002.