



**HAL**  
open science

# Heuristiques d'ordonnement pour une résolution embarquée de conflits aériens par une méthode séquentielle

Nicolas Archambault, Géraud Granger, Nicolas Durand

► **To cite this version:**

Nicolas Archambault, Géraud Granger, Nicolas Durand. Heuristiques d'ordonnement pour une résolution embarquée de conflits aériens par une méthode séquentielle. RIVF 2004, 2ème Conference Internationale Associant Chercheurs Vietnamiens et Francophones en Informatique, Feb 2004, Hanoi, Vietnam. pp 1-6. hal-00938234

**HAL Id: hal-00938234**

**<https://enac.hal.science/hal-00938234>**

Submitted on 25 Apr 2014

**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.

# Heuristiques d’ordonnancement pour une résolution embarquée de conflits aériens par une méthode séquentielle

Nicolas Archambault, Géraud Granger and Nicolas Durand  
email : (archamba,durand,granger)@recherche.enac.fr

**Abstract—** The resolution of conflicts between  $n$  aircraft is highly combinatorial and cannot be optimally solved using classical mathematical optimisation techniques. Using a priority order to solve a  $n$ -aircraft conflict is much easier but the solution is not optimal. However it is difficult to determine the best order, or even an order that is good enough to ensure that a solution exists. In this paper, a comparative study on a set of proposed heuristics to provide a suitable order is presented. Heuristics are tested on simulations using real traffic data.

## I. INTRODUCTION

### A. La gestion du trafic aérien

La gestion de trafic aérien (ATM, pour Air Traffic Management) a pour objectif de permettre aux aéronefs de réaliser leur vol selon leurs préférences (en matière d’horaires, de vitesse, de niveau de vol) tout en assurant leur sécurité. En particulier, le contrôle du trafic aérien (ATC, pour Air Traffic Control) assure la séparation entre aéronefs au cours de leur vol. Une situation où les normes de séparations entre aéronefs sont potentiellement violées est appelée “conflit”. La détection et la résolution des conflits font partie des missions du contrôle aérien. La conception d’outils fiables d’aide à la détection et à la résolution pour le contrôleur aérien répondent à l’enjeu de l’augmentation de la capacité du contrôle et de la capacité de l’espace aérien en terme de nombre de vols.

### B. Résolution de conflits

La résolution de conflit repose généralement sur l’exécution par un ou plusieurs aéronefs de manœuvres d’évitement : en général, manœuvres de déviation dans le plan horizontal, dans le plan vertical, ou modification de la vitesse de croisière. Ces manœuvres sont similaires à celles ordonnées par les contrôleurs aériens.

Diverses solutions algorithmiques ont été proposées pour résoudre des situations de conflit aérien, employant diverses catégories de méthodes : optimisation par algorithmes génétiques [1] [2], contrôle optimal, systèmes multi-agents [3] [4] [5], programmation semi-définie [6], modélisation des avions comme des particules chargées dans un champ de forces répulsives [7].

N. Archambault - Coframi

G. Granger - Steria Département ATS

N. Durand - LOG - Centre d’Études de la Navigation Aérienne ; 7, avenue Édouard Belin, 31055 Toulouse cedex 4, FRANCE

## II. CADRE

### A. Objectif

Dans cette étude, nous proposons d’utiliser une méthode de résolution séquentielle en 1 contre  $n$  pour résoudre des conflits aériens impliquant plusieurs aéronefs. La résolution utilise un algorithme  $A^*$ .

Nous étudions la faisabilité d’une résolution basée sur cette méthode et insistons plus particulièrement sur l’influence de l’ordonnancement pour l’efficacité de cette méthode de résolution séquentielle.

### B. Contexte

Le centre expérimental Eurocontrol (CEE) a développé au début des années 90 le projet FreeR [12], [13], perspective d’un nouveau système de contrôle aérien où la séparation entre aéronefs (c’est-à-dire le respect de distances inter-aéronefs normatives dites de séparation) est assurée, dans certains espaces aériens, non plus par des systèmes et contrôleurs basés au sol mais par des systèmes embarqués.

Dans le contexte “Free-Flight”, les aéronefs suivaient des routes rectilignes entre leur origine et leur destination, et non des routes survolant des balises.

Aujourd’hui, deux nouveaux concepts novateurs (SuperSector et SectorLess) [14] réfléchissent à une nouvelle organisation de l’espace aérien, dans laquelle les procédures actuelles de contrôle devront être radicalement redéfinies. On peut ainsi envisager une délégation partielle aux aéronefs ou totale du maintien de séparation. Ce mode de contrôle ferait lui aussi appel à des algorithmes de résolution autonome, et c’est dans ce contexte que se place cette étude.

### C. Principe

Le principe de la méthode est une résolution séquentielle sur un modèle distribué embarqué, c’est-à-dire que, un par un, chaque avion concerné va calculer lui-même une trajectoire optimale libre de conflits. Nous supposons qu’en deçà d’une certaine distance, les avions sont capables de détecter les autres avions, et de leur communiquer une prévision de trajectoire et des informations supplémentaires que nous détaillerons (voir IV) pour déterminer un ordre de priorité entre eux.

L’avion le plus prioritaire déterminera pour lui-même une trajectoire optimale sans contrainte et la communiquera aux

autres avions, puis chaque avion dans l'ordre de priorité déterminera pour lui-même une telle trajectoire libre de conflits en tenant compte comme contraintes des trajectoires des avions ayant déjà déterminé leur trajectoire.

Le critère d'efficacité sur les solutions libres de conflits est la longueur des trajectoires. Plus les déviations sont courtes, plus le surcoût en consommation et le retard occasionné sont faibles.

### III. DÉFINITIONS

#### A. Conflits et clusters

On définit une fenêtre d'anticipation  $F_a$  et un temps de rafraîchissement  $\Delta$ . La détection de conflit est faite toutes les  $\Delta$  minutes pour les  $F_a$  prochaines minutes. Pendant le temps  $\Delta$ , le système doit effectuer la détection de conflits, procéder à l'échange d'informations entre les avions et proposer à chaque avion une manœuvre sans conflit.

On dit que deux avions sont en conflit lorsqu'à un instant de la situation anticipée, leur différence d'altitude est inférieure à une norme de séparation verticale et que simultanément la distance entre leurs projections dans le plan horizontal est inférieure à une norme de séparation horizontale.

La détection de conflits se fait donc par paires d'avions, en tenant compte du modèle d'incertitudes dans le plan horizontal et dans le plan vertical. En routes directes, il s'agit donc de mesurer dans le plan horizontal la distance entre deux segments, et ce pour chaque pas de temps. Dans le plan vertical, on compare le niveau bas de l'avion le plus haut avec le niveau haut de l'avion le plus bas.

Le rayon de détection doit être suffisamment important pour assurer que deux avions qui seraient en conflit au cours des  $F_a$  prochaines minutes seront identifiés comme étant en conflit.

A partir des paires d'avions en conflit, les clusters à résoudre sont identifiés par fermeture transitive de la relation "est en conflit avec". Notre algorithme cherche à résoudre les conflits au sein de chaque cluster.

Il est important de constater qu'il n'existe pas forcément un instant où tous les avions du cluster sont en conflit simultanément. Une telle définition de la relation de conflit conduit à résoudre des clusters impliquant peu d'avions. Ici la détection des conflits sur la fenêtre temporelle  $F_a$  conduit dans les simulations à résoudre fréquemment des clusters de plus de 10 aéronefs.

#### B. Manœuvres

Les manœuvres d'évitement doivent être simples à réaliser par un pilote humain, et faciles à comprendre par les pilotes des avions voisins concernés ; nous nous inspirons donc fortement des manœuvres déjà utilisées dans le cadre du contrôle actuel.

Nous nous limitons à envisager pour les avions deux types de manœuvres d'évitement : des manœuvres de déviation dans le plan horizontal et des manœuvres dans le plan vertical. Les manœuvres du premier type sont de type point tournant : l'avion altère son cap de 10, 20 ou 30 degrés à droite ou à gauche de sa trajectoire à  $t_0$  et reprend un cap vers sa destination à  $t_1$  (voir figure 1). Ce type de manœuvre permet de rester proche de la trajectoire optimale [2]. Les manœuvres du deuxième type dépendent de la phase de vol de l'avion (voir figure 2). Ainsi

pour un avion stable, la manœuvre consiste à amorcer sa descente vers le niveau de vol inférieur au temps  $t_0$  et d'amorcer sa remontée au temps  $t_1$ .

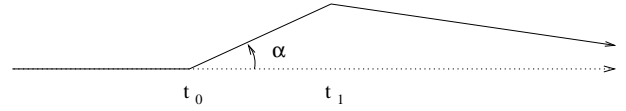


Fig. 1. Manœuvre dans le plan horizontal.

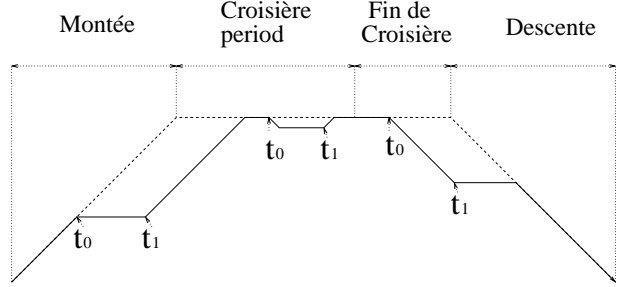


Fig. 2. Manœuvres dans le plan vertical.

Les manœuvres horizontales et verticales sont donc décrites respectivement par trois ( $t_0$ ,  $t_1$ , et l'angle) et par deux variables, ce qui permet de limiter l'espace de recherche.

#### C. Algorithme de résolution

La résolution proprement dite consiste à trouver pour un avion une trajectoire libre de conflit en considérant  $n$  autres trajectoires comme contraintes. L'espace des solutions est décrit par trois variables,  $t_0$ ,  $t_1$  et la modification de trajectoire ; cette dernière consiste en un angle de déviation  $\alpha$  ou en une manœuvre de changement de niveau. Pour rechercher et évaluer dans cet espace de solutions, la meilleure trajectoire, le solveur embarqué utilise un algorithme de type  $A^*$ , dont on trouvera une description dans [15].

Au cours de la résolution, l'avion peut passer par 4 états successifs. Dans le premier état ( $E_0$ ), l'avion suit sa trajectoire prévue jusqu'au temps  $t_0$ . Dans le deuxième état ( $E_1$ ), à partir de  $t_0$ , l'avion est dévié de sa trajectoire d'origine de plus ou moins 10, 20 ou 30 degrés. Ce cap est conservé jusqu'au temps  $t_1$ . On notera  $E_1V$  l'état correspondant à une manœuvre verticale. Dans le troisième état ( $E_2$ ), au temps  $t_1$ , l'avion prend le cap qui lui permet de rejoindre directement sa position finale. Enfin dans le dernier état ( $E_3$ ), l'avion a atteint sa position finale.

On appellera par la suite *avion central* l'avion qui cherche à résoudre et *avions contraintes* ceux qui ont déjà résolu.

La nouvelle trajectoire de l'avion est donc entièrement déterminée par  $t_0$ ,  $t_1$  et le cap pris en déviation.

– Tant qu'un avion est dans l'état  $E_0$ , il choisit à chaque pas de temps entre 8 actions possibles :

- 1) Conserver son cap d'origine, en restant dans la première étape de l'évitement ( $E_0$ ) ;
- 2) Modifier sa trajectoire en opérant un changement de cap de 10, 20 ou 30 degrés à droite ou 10, 20, ou 30 degrés à gauche, et ainsi passer dans le deuxième état de l'évitement ( $E_1$ ) ;

- 3) Amorcer une manœuvre verticale, dont la nature dépend de sa phase de vol ( $E_1V$ ) : interruption de montée, anticipation de descente, ou changement de niveau de vol en croisière.
- Lorsqu’un avion est dans l’état  $E_1$ , il a le choix entre 2 actions possibles :
    - 1) Poursuivre sa route en conservant son cap modifié, en restant dans l’état ( $E_1$ ) ;
    - 2) Opérer un virage pour suivre un cap direct vers sa position finale, il passe alors dans l’état ( $E_2$ ).
  - Lorsque l’avion est dans l’état  $E_1V$  (comme dans le cas  $E_1$ ), il a le choix entre 2 actions possibles :
    - 1) Poursuivre sa manœuvre verticale, en restant dans l’état ( $E_1V$ ) ;
    - 2) Terminer sa manœuvre d’évitement, il passe alors dans l’état ( $E_2$ ).
  - Un avion dans l’état  $E_2$  ne change pas d’état, sauf s’il a atteint sa destination (il passe alors dans l’état  $E_3$ ).

Les temps  $t_0$  et  $t_1$  prennent des valeurs discrètes (multiples du pas de temps). Les trajectoires d’évitement ainsi définies sont représentées par les chemins d’un arbre.

La fonction de coût  $k(u, v)$  du parcours de l’arbre calcule la distance, pour l’avion central uniquement, entre les positions au nœud  $u$  et au nœud  $v$ . Dans le cas d’une manœuvre vertical, ce coût est augmenté, parce que la manœuvre verticale pénalise l’avion (en termes de consommation et de temps de vol) sans considérablement rallonger sa trajectoire. La fonction de coût tient aussi compte des séparations, et si celles-ci ne sont pas respectées dans l’état  $v$  alors la fonction  $k(u, v)$  retourne une valeur très importante qui stoppe l’évaluation de la branche dans cette direction.

L’heuristique sur laquelle repose l’algorithme  $A^*$  doit donner une estimation du coût du nœud courant à un état terminal. La distance entre les positions prises par l’avion central dans le nœud courant et dans l’état terminal minore le coût entre ces deux nœuds (toute autre trajectoire est plus longue que la ligne droite). Cette heuristique, minorante, possède donc la propriété intéressante de permettre à l’algorithme  $A^*$  de toujours trouver la solution optimale.

#### D. Algorithme d’ordonnancement

L’algorithme qui, à partir d’un ordre total défini sur les avions d’un cluster (voir IV) détermine l’ordre dans lequel les avions décident de leur trajectoire, repose sur le principe d’une distribution de jetons et est décrit dans [17]. Nous le rappelons ici.

Chaque avion a la connaissance des avions en conflit qui sont dans son rayon de détection. Il sait, pour ces avions et pour ces avions seulement, lesquels sont plus prioritaires et lesquels sont moins prioritaires que lui. Pour pallier à ce problème de la visibilité partielle, un système de distribution de jetons est utilisé, permettant à chaque avion de savoir si son tour est venu de choisir sa trajectoire, ou s’il doit encore attendre que d’autres avions plus prioritaires que lui optimisent d’abord la leur.

Chaque avion distribue un jeton aux avions présents dans son rayon de détection qui sont en conflit avec lui-même ou avec tout autre avion, et qui sont moins prioritaires que lui (les avions

n’étant pas en conflit ne reçoivent jamais de jeton). Cette distribution initiale de jetons, préalable à la résolution des conflits du cluster, permet ensuite d’appliquer la règle suivante : si un avion n’a pas de jeton, il peut déterminer une trajectoire optimale respectant les contraintes de séparation avec tous les avions de son rayon de détection n’ayant pas de jeton eux-mêmes en ne tenant pas compte des avions ayant des jetons. Une fois cette opération réalisée, il reprend tous les jetons qu’il a donnés. Cette règle est appliquée jusqu’à ce que plus aucun avion n’ait de jeton.

On montre dans [18] que cette technique de *distribution / résolution / récupération* de jetons ne peut pas entraîner de situation de blocage du processus de résolution. On ne peut pas se trouver dans une situation où tous les avions auraient au moins un jeton, ni dans une situation où deux avions en conflit sans jeton devraient résoudre simultanément.

## IV. ORDONNANCEMENT

### A. Principe

L’idée d’un ordonnancement des avions par priorité est déjà appliquée pour l’évitement des aéronefs en vol à vue (VFR, pour Visual Flight Rules) ; l’extension des règles VFR dans un contexte de “Free-Flight” est abordée dans [19]. Dans [20], la recherche d’un ordonnancement établit un parallèle entre l’ordonnancement pour la résolution de conflit et l’ordonnancement de tâches.

L’ordonnancement des avions dans la résolution est un facteur crucial pour l’efficacité de la résolution. En effet, même avec un nombre d’avions faible, l’algorithme peut aboutir à une solution efficace, peu efficace, ou ne pas aboutir à une solution.

Nous présentons ici une méthode pour obtenir un ordonnancement efficace (c’est-à-dire qui conduit à une solution efficace). Cette méthode repose sur l’utilisation d’heuristiques appropriées. Ces heuristiques doivent permettre qu’un avion dont la résolution est potentiellement plus difficile soit plus prioritaire et résolve ses conflits avant les autres, plutôt qu’après. En effet, un avion qui décide tard est limité par les trajectoires des avions qui sont plus prioritaires que lui.

On peut envisager plusieurs heuristiques pour caractériser a priori les difficultés que rencontrera un avion dans sa résolution : nombre de conflits directs où il est impliqué au sein du cluster, urgence de son premier conflit, manœuvrabilité intrinsèque...

Afin d’obtenir un ordre total sur les avions d’un cluster, on combine les heuristiques pour réaliser un ordre pseudo-lexicographique. Lorsque qu’un critère heuristique ne permet pas de comparer deux avions (parce qu’ils ont le même nombre de conflits par exemple), on utilise successivement les autres critères jusqu’à ce que l’un des critères distingue les deux avions. On termine toujours par le “numéro CAUTRA” des avions, qui est une immatriculation unique et assure un ordre total.

### B. Notations

On utilisera les notation suivantes pour désigner les critères heuristiques :

- 
- $m$  la manœuvrabilité de l’avion ;
  - $f$  l’instant du premier conflit ;

$n$  le nombre de conflits qui impliquent l’avion dans la fenêtre d’anticipation ;

$i$  le numéro CAUTRA.

Si la lettre est minuscule ( $f, m, n, i$ ), l’avion le plus prioritaire sera celui pour lequel la grandeur comparée est la plus petite ; si elle est majuscule ( $F, M, N, I$ ), celui pour lequel elle est la plus grande.

Par exemple pour l’heuristique “ $MfNi$ ”, l’avion 1 est plus prioritaire que l’avion 2 :

- si son nombre de manœuvres déjà prévu est plus grand (il est donc moins manœuvrable) ;
- en cas d’égalité pour le critère précédent : si son instant de premier conflit est plus petit (conflit urgent, rapproché) ;
- en cas d’égalité pour les critères précédents (par exemple s’il s’agit des deux premiers avions en conflit, et qu’ils n’ont pas de manœuvres déjà prévues) : si son nombre de conflits (avec les autres avions du cluster) est plus grand ;
- en cas d’égalité pour les critères précédents : si son numéro CAUTRA est plus petit (on est assuré que les numéros des deux avions sont distincts).

### C. Méthode expérimentale

Les heuristiques sont d’abord testées sur un exemple “jouet” (avions volant à même altitude et même vitesse dans un espace restreint, point d’origine et de destination aléatoires). On crée aléatoirement un certain nombre de ces situations et on compare les résultats obtenus par les différentes heuristiques.

Les critères d’évaluation des résultats sont les suivants :

- L’ordonnancement conduit-il à une solution sans conflit ? C’est évidemment le critère principal.
- Si on n’a pas de solution globale, combien d’avions ont trouvé une solution respectant les contraintes des trajectoires des avions ayant déjà résolu ?
- Quel est le retard causé par les manœuvres pour éviter le conflit (pour les avions ayant résolu tous leurs conflits) ?

On constate que les heuristiques ont des performances très éloignées les unes des autres. Et également que pour toute paire d’heuristiques testées dans ces situations artificielles, il existe des situations dans laquelle une seule des deux fournit un ordonnancement qui permet de résoudre tous les conflits du cluster.

Puis on teste l’algorithme de résolution sur plusieurs journées de trafic. Les données utilisées sont les plans de vol déposés, les avions volent en route directe. La détection de conflits se fait dans l’espace supérieur, espace aérien concerné par le concept de “Free-Flight” en raison de la relativement faible densité de trafic. On peut aussi supposer que l’hypothèse de faible densité restera valable dans le concept SuperSector/SectorLess, en raison d’une meilleure répartition du trafic.

La comparaison ne se fait plus en agglomérant les comparaisons sur chaque situation, mais de façon directement globale. Plus précisément, après les premiers tests, on constate que les heuristiques testées ne permettent pas de résoudre tous les conflits qui se présentent. Le critère de comparaison sera donc d’abord le nombre de conflits non résolus, puis éventuellement des critères secondaires d’efficacité des manœuvres. Il s’agit là d’un parti pris. En effet on aurait pu simuler plusieurs journées de trafic avec détection mais sans résolution de conflit, et en

extraire tous les conflits réels, afin de tester artificiellement les heuristiques sur le même ensemble de conflits réels. Cependant l’objectif final est d’avoir un système qui fonctionne en continu, sur des journées ininterrompues de trafic, c’est-à-dire dans les conditions normales d’utilisation du simulateur, où la résolution des conflits se répercute sur le reste de la simulation, et c’est sur ce type de simulation que nous comparons les heuristiques. Les tests sont effectués sur le simulateur arithmétique de trafic CATS [21].

### D. Résultats

Nous présentons ici les résultats obtenus pour des heuristiques caractéristiques sur un échantillon de journées de trafic. On compare les heuristiques et on mesure l’apport des manœuvres verticales. Pour chaque date, figurent les résultats des résolutions sans manœuvres verticales (H) et avec les manœuvres verticales (V).

	24/01		26/01		02/02		04/03	
	317		372		468		368	
	H	V	H	V	H	V	H	V
$fMNi$	9	1	21	0	32	1	14	1
$MNi$	10	1	21	0	31	5	13	1
$NfMi$	7	3	21	0	33	3	11	1
$fNmi$	7	3	21	0	33	3	11	1
$fMni$	10	1	21	0	35	1	12	3
$MfNi$	9	1	21	0	32	3	13	1
$fNmi$	23	9	51	14	74	30	27	14

TABLE I  
NOMBRE DE CONFLITS NON RÉSOLUS

	24/01		26/01		02/02		04/03	
	317		372		468		368	
	H	V	H	V	H	V	H	V
$fMNi$	523	518	605	612	883	901	668	652
$MNi$	519	514	602	612	882	895	676	656
$NfMi$	530	515	599	610	890	901	656	661
$fNmi$	530	515	599	610	890	901	656	661
$fMni$	522	516	607	615	885	900	655	654
$MfNi$	526	520	602	612	883	897	676	658
$fNmi$	402	407	464	486	668	677	517	521

TABLE II  
NOMBRE TOTAL DE MANŒUVRES

L’analyse des résultats montre que l’utilisation de manœuvres verticales en plus des manœuvres horizontales permet de résoudre la quasi-totalité des conflits (la proportion de conflits non résolus, de 3 - 9 %, passe à 0,3 - 1 %) (tableau I). Pourtant le nombre de manœuvres nécessaires pour résoudre les conflits (tableau II) n’est pas nécessairement plus important, ni les manœuvres et les délais qui en découlent plus longs (tableau III). Une piste à explorer est donc la diversification des

	24/01		26/01		02/02		04/03	
	317		372		468		368	
	H	V	H	V	H	V	H	V
<i>fMNi</i>	73	74	81	78	84	77	73	73
<i>MNi</i>	73	74	82	77	84	74	73	74
<i>NfMi</i>	73	74	81	77	84	77	75	73
<i>fNMI</i>	73	74	81	77	84	77	75	73
<i>fMni</i>	74	73	81	77	84	78	74	74
<i>MfNi</i>	72	74	82	77	84	74	73	73
<i>fNmi</i>	92	101	111	99	106	112	92	94

TABLE III  
DURÉE MOYENNE D'UNE MANŒUVRE

manœuvres, en autorisant par exemple les modifications de vitesse de l'aéronef.

On constate également que, quand bien même certaines heuristiques sont nettement inefficaces (ainsi l'heuristique *fNmi* échoue à résoudre un nombre relativement important de conflits), qu'il est difficile même sur un nombre réduit de simulations de dégager une heuristique invariablement meilleure. De plus les performances relatives des heuristiques dépendent également des manœuvres possibles. Sur une journée donnée, la meilleure heuristique n'est pas la même dans les deux cas (H et V).

Cependant les simulations montrent que l'heuristique élémentaire *M*, qui consiste à favoriser les avions déjà manœuvrants et donc moins susceptibles de modifier leur trajectoire, est tout-à-fait indiquée, dans le cadre de nos hypothèses de simulation.

Si tous les conflits rencontrés dans une journée de trafic ne sont pas résolus, les rares conflits non résolus peuvent être étudiés individuellement afin de mettre en évidence les mécanismes qui rendent un cluster difficile.

### E. Inversion de priorité

L'analyse des conflits non résolus par l'algorithme met souvent en évidence une situation d'inversion locale de priorité. Ainsi, lors d'une première itération de résolution, supposons que l'avion *A* soit plus prioritaire que l'avion *B*. L'avion *B* déterminera donc sa trajectoire en deuxième position, en tenant compte de la trajectoire de l'avion *A*. Supposons que la manœuvre d'évitement qu'il opère dure plus d'un pas de temps. A l'itération suivante, rien n'empêche dans notre modèle que l'avion manœuvrant *B* devienne plus prioritaire. Ce sera en particulier le cas pour un ordonnancement selon l'heuristique *M*, simplement parce que *B* est manœuvrant et pas *A*. *B* doit donc optimiser sa trajectoire sans tenir compte de l'avion *A*. Il est donc envisageable qu'il reconsidère la manœuvre qu'il effectuait, si celle-ci lui était imposée par la présence de l'avion *A*. L'avion *A* doit également reconsidérer sa manœuvre puisque *B* a maintenant résolu avant lui. Dans un premier temps, afin de réduire les conséquences d'une telle inversion de priorité, nous avons limité les possibilités d'un avion manœuvrant de reconsidérer sa manœuvre : dans notre manœuvre en point tournant, d'une itération sur l'autre, un avion ne peut que prolonger ou

raccourcir la durée du changement de cap. Nous avons dans un deuxième temps interdit qu'une manœuvre soit raccourcie.

Une autre approche consiste à éviter l'inversion de priorité dans les cas problématiques. Dans cette approche, les avions manœuvrants gardent en mémoire les avions plus prioritaires qu'eux qui les ont contraint à manœuvrer. Mais le problème de la transitivité d'un ordre de priorité reposant sur cette méthode se pose immédiatement ; nous l'illustrons dans un cas à 4 avions. Supposons dans un premier temps deux couples d'avions en conflit  $A_1 \succ A_2$ ,  $B_1 \succ B_2$ . Les deux clusters sont indépendants.  $A_2$  et  $B_2$ , moins prioritaires, manœuvrent. On souhaite qu'à l'itération suivante les priorités ne soient pas inversées.

Supposons qu'à l'itération suivante les visibilitées soient telles qu'indiquées sur la figure 3 (les arcs représentent les portées de visibilité et les flèches les priorités). Les 4 avions constituent un seul cluster. Du fait de la mémoire des priorités introduite ici on a toujours  $A_1 \succ A_2$ ,  $B_1 \succ B_2$ . On ne peut pas exclure que les priorités déterminées localement indépendamment dans les autres paires en visibilité donnent  $B_2 \succ A_1$  et  $A_2 \succ B_1$ . En effet, les critères utilisés pour déterminer les priorités à la deuxième itération (manœuvrabilité, nombre de conflits, instant du premier conflit) ne prennent pas forcément les mêmes valeurs que celles utilisées à l'itération précédente (et qui sont toujours utilisées du fait de la mémoire, partielle, des priorités). Or la transitivité de l'ordre est ici violée.

Le maintien de l'ordre de priorité entre les avions manœuvrants et les avions avec lesquels ils sont en conflit ne se satisfait donc pas du cadre de l'algorithme de détermination des priorités. En effet celui-ci suppose que l'échange d'informations préalable à la résolution ne demande qu'une diffusion unique par chaque avion de son état, de sa prévision de trajectoire et de son environnement, en direction des avions à sa portée. La prise en compte de priorités héritées du passé demande une négociation en plusieurs phases afin de s'assurer de la transitivité de l'ordre de priorité défini.

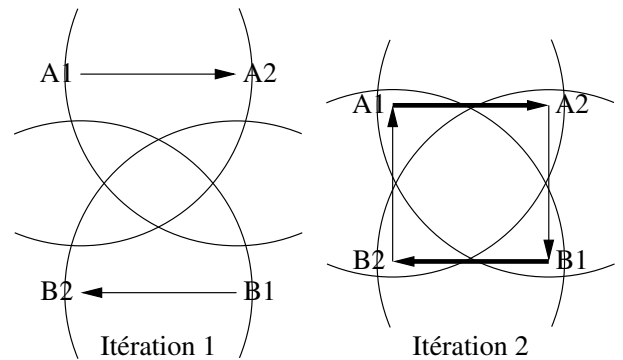


Fig. 3. Inversion de priorité.

## V. CONCLUSION

Ces résultats montrent l'importance de l'ordonnement dans la résolution séquentielle de clusters d'avions en conflit. En effet la performance du solveur est considérablement modifiée en fonction de l'heuristique utilisée. Les heuristiques

ne sont pas aisément comparables deux à deux, mais les heuristiques combinant la manœuvrabilité des aéronefs et leurs contraintes d'évolution (nombre de conflits à résoudre, urgence du conflit) sont relativement efficaces.

---

#### REFERENCES

- [1] J. M. Alliot, Hervé Gruber, Marc Schoenauer, "Using genetic algorithms for solving ATC conflicts" in *Proceedings of the Ninth Conference on Artificial Intelligence Application*. IEEE, 1993.
  - [2] N. Durand, "Optimisation de trajectoires pour la résolution de conflits en-route", Thèse de doctorat, INPT, 1996.
  - [3] C. Tomlin, G. J. Pappas, S. Sastry, "Conflict resolution for Air Traffic Management : a case study in multi-agent hybrid systems", University of California at Berkeley, 1996.
  - [4] J. P. Wangermann, R. F. Stengel, "Optimization and coordination of multi-agent systems using principled negotiation" in *Proceedings of AIAA Guidance, Navigation, and Control Conference*, San Diego, 1996.
  - [5] C. Goodchild, M. A. Vilaplana, S. Elefante, "Cooperative optimal airborne separation assurance in Free-Flight airspace", *3rd Usa/Europe ATM Seminar*, Napoli, 2000.
  - [6] J. H. Oh, E. Frazzoli, Z. H. Mao, E. Feron, "Resolution of conflicts involving many aircraft via semidefinite programming.", Technical report, MIT, 1999.
  - [7] K. Zeghal, "A review of different approaches based on force fields for airborne conflict resolution" in *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Boston, 1998.
  - [8] M. Prandini, J. Hu, J. Lygeros, S. Sastry, "Aircraft conflict prediction and resolution using brownian motion" in *Proceedings of the 38st Conference on Decision and Control*, Phoenix, 1999.
  - [9] J. Kosecka, C. Tomlin, G. Pappas, S. Sastry, "Generation of conflict resolution maneuvers for Air Traffic Management", International Conference on Robotics and Intelligent Systems, 1997.
  - [10] P. K. Menon, G. D. Sweriduk, "Optimal strategies for Free-Flight Air Traffic conflict resolution", *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 2, March-April 1999.
  - [11] M. S. Eby, "A self-organizational approach for resolving Air Traffic Conflicts", *Lincoln Laboratory Journal*, Vol. 7, No. 2, 1994.
  - [12] V. Duong, E. Hoffman, J. P. Nicolaon, "Initial results of investigation into autonomous aircraft concept (FREER-1)" in *Proceedings of the 1st USA/Europe ATM Seminar*, 1997.
  - [13] V. Duong, E. Hoffman, "Conflict resolution advisory in autonomous aircraft operations" in *Proceedings of the 16th AIAA/IEEE Digital Avionics Systems Conference*, 1997.
  - [14] V. Duong, G. Gawinowski, J. P. Nicolaon, D. Smith, "Sector-Less Air Traffic Management" in *Proceedings of the Air Traffic Management Conference*, Santa Fe, 2001.
  - [15] J. M. Alliot, T. Schiex, P. Brisset, F. Garcia, *Intelligence Artificielle et Informatique Théorique* Cepadues, 2002.
  - [16] N. Durand, J. M. Alliot, "Optimal resolution of en-route conflicts" in *Proceedings of the 1st USA/Europe ATM Seminar*, 1997.
  - [17] G. Granger, N. Durand, J. M. Alliot, "Token allocation strategy for Free-Flight conflict solving", IJCAI, Seattle, 2001.
  - [18] G. Granger, "Détection et résolution de conflits aériens : modélisations et analyse", Thèse de doctorat, INPT, 2002.
  - [19] V. Duong, E. Hoffman, F. Floc'hic, J. P. Nicolaon, A. Bossu, "Extended Flight Rules (EFR) to apply to the resolution of encounters in autonomous airborne separation", Technical Report, Eurocontrol, 1996.
  - [20] J. Carlier, V. Duong, D. Nace, H. H. Nguyen, "Using disjunctive scheduling for optimally solving en-route conflicts", AIAA/IEEE 21st Digital Avionics Systems Conference, Irvine, 2002.
  - [21] J. M. Alliot, N. Durand, J. Bosc, L. Maugis, "Cats : a complete air traffic simulator" in *Proceedings of the 16th AIAA/IEEE Digital Avionics Systems Conference*, 1997.
-