



HAL
open science

Optimisation de trajectoires pour la résolution de conflits aériens en route

Nicolas Durand

► **To cite this version:**

Nicolas Durand. Optimisation de trajectoires pour la résolution de conflits aériens en route. Optimisation et contrôle [math.Oc]. INPT, 1996. Français. NNT : 1996INPT033 . tel-01293710v1

HAL Id: tel-01293710

<https://theses.hal.science/tel-01293710v1>

Submitted on 29 Mar 2016 (v1), last revised 6 Mar 2023 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Thèse de Doctorat

présentée en vue de l'obtention du titre de

Docteur de l'Institut National Polytechnique de Toulouse

Spécialité : **INFORMATIQUE**

par

Nicolas Durand

Optimisation de trajectoires pour la résolution de conflits aériens en route

Le 28 mai 1996

Composition du Jury :

M. Jean-Marc ALLIOT - Examineur
M. Pierre BOURGINE - Rapporteur
M. Dominique COLIN DE VERDIERE - Examineur
M. Joseph NOAILLES - Directeur de Thèse
M. Gérard PADIOU - Examineur
M. Philippe TOINT - Rapporteur

Résumé

L'augmentation du trafic aérien génère des problèmes d'optimisation très complexes pour le Contrôle Aérien. Le problème d'optimisation de trajectoires pour la résolution de conflits en route reste aujourd'hui ouvert. Dans cette thèse, l'approche initiale est d'étudier le problème de conflit à deux avions comme un problème de commande optimale avec contrainte sur l'état. Cette approche permet de dégager certaines propriétés des trajectoires optimales. L'étude théorique de la structure de l'ensemble des trajectoires admissibles fait apparaître le caractère fortement combinatoire du problème. Des applications numériques effectuées avec un algorithme de type gradient permettent alors d'illustrer les résultats théoriques et de constater l'inefficacité des méthodes locales. Dans un deuxième temps, les résultats théoriques sont exploités afin de définir les premiers éléments de modélisation du problème réel permettant de tenir compte des contraintes opérationnelles liées au contexte du contrôle du trafic aérien. Le caractère global et fortement combinatoire du problème d'optimisation conduit à utiliser les algorithmes génétiques. Ceux-ci permettent d'approcher plusieurs solutions différentes proches de l'optimum. Un opérateur de croisement adapté aux fonctions partiellement séparables est introduit dans les algorithmes génétiques de manière à pouvoir résoudre des conflits pouvant impliquer jusqu'à une vingtaine d'avions en temps réel avec de faibles populations. Les résultats sont illustrés par plusieurs exemples.

Mots clés :

Contrôle du trafic aérien, résolution de conflits, algorithmes génétiques, séparabilité partielle, optimisation combinatoire, optimisation global.

Abstract

As air traffic keeps increasing, the resolution of En Route conflicts is becoming a serious concern, especially in Europe. In this thesis, the two aircraft conflict problem is first studied as an optimal command problem with state constraints : this study enlightens some interesting properties of the optimal trajectories. The structure of the admissible trajectories set makes the problem combinatorial. Numerical tests run with local optimization tools confirm that local optimization algorithms are not suited to solve such problems. The theoretical results are then used to build a model that can handle many aircraft and Air Traffic Control real constraints. As the optimization problem is global and combinatorial, Genetic Algorithms are suited to it. Furthermore, Genetic Algorithms find several local optima near the global optimum. Genetic operators adapted to partially separable functions are introduced in order to improve the algorithms efficiency on conflicts problems involving up to 20 aircraft.

Keywords :

Air traffic control, conflict solving, genetic algorithms, partial separability, combinatorial optimization, global optimization.

Remerciements

J'adresse de sincères remerciements aux membres du jury qui ont accepté de juger ce travail.

- Monsieur Paul Bourgine du CREA à l'Ecole Polytechnique, qui m'a fait l'honneur d'être rapporteur de ces travaux.
- Monsieur Philippe Toint, professeur à l'Université de Namur, qui m'a fait l'honneur d'être rapporteur de ces travaux.
- Monsieur Gérard Padiou, professeur à l'ENSEEIH, pour l'honneur qu'il me fait d'être examinateur.
- Monsieur Dominique Colin de Verdière, chef adjoint du CENA, qui m'a fait confiance dans l'orientation de mes travaux auxquels il a toujours porté un grand intérêt.
- Monsieur Joseph Noailles, mon directeur de thèse, pour m'avoir fait confiance en me proposant cette thèse, pour ses précieux conseils, sa disponibilité et que j'espère n'avoir pas déçu.
- Jean-Marc Alliot, qui a toujours été présent, et m'a continuellement soutenu et guidé durant trois ans.
- Monsieur Xavier Fron, du Centre Expérimental Eurocontrol de Brétigny, membre invité du jury.

Mes remerciements vont également aux personnes du laboratoire Informatique et Mathématiques Appliquées de l'ENSEEIH, et notamment l'équipe Algorithmique Parallèle et Optimisation du deuxième étage pour leur aide durant ma première année de thèse et pour la bonne ambiance qui règne au deuxième étage. Je remercie particulièrement Jean-Yves L'Excellent et Samuel Gruska avec qui j'ai partagé le bureau, et qui m'ont beaucoup aidé à commencer mes travaux.

Je tiens à exprimer mes remerciements aux personnes avec qui j'ai pu travailler à l'ENAC, Nicolas Alech, Frédéric Médioni, Olivier Chansou, Jean-Baptiste Gottland, Yann Lefablec, Daniel Delahaye et Jean-François Bosc, sans oublier Christophe Bontemps et Isabelle Alliot.

J'exprime ma reconnaissance à Stéphane Chatty, Ruby et Laurence Garcia pour leur aide dans mes démarches administratives.

Enfin, je remercie Monsieur Jean-Marc Garot, Directeur du centre expérimental EUROCONTROL de Brétigny, de m'avoir permis de faire cette thèse au CENA, et de m'avoir fait confiance dans les orientations prises.

Table des matières

1	Le contrôle du trafic aérien	13
1.1	Définitions	13
1.2	Contexte du problème	15
1.3	Les projets d'automatisation existants	18
1.3.1	La tentative américaine AERA	18
1.3.2	ARC2000 et ses dérivés	21
1.3.3	Le projet SAINTEX	22
1.3.4	Les techniques réactives	23
1.3.5	Conclusions	25
1.4	Hypothèses de travail	27
1.4.1	Premières spécifications	27
1.4.2	Contraintes opérationnelles	28
2	Modélisation par Contrôle Optimal	31
2.1	Résolution de conflits : problème de Contrôle Optimal	32
2.1.1	Conflit entre deux avions non contraints en vitesse	32
2.1.2	Analyse de la sensibilité aux paramètres géométriques	40
2.2	Le cas de deux avions lorsque l'un est privilégié.	44
2.2.1	Structure de l'ensemble des trajectoires admissibles	44
2.2.2	Trajectoires optimales de l'avion dévié	47
2.2.3	Influence de l'angle d'incidence	52
2.3	Les limites du modèle pour $n > 2$ avions	56
2.4	Approche numérique	57
2.4.1	Conflit à deux avions	57
2.4.2	Conflit à 5 avions	57
2.4.3	Remarques	59
3	Modélisations simplifiées : point tournant et "offset"	65
3.1	Approximation par point tournant	65
3.1.1	Calcul préliminaire	66
3.1.2	Représentation graphique	67
3.1.3	Influence de l'angle d'incidence	67
3.2	Approximation par offset	70

3.2.1	Définition de l'offset	70
3.2.2	Problème linéaire avec contraintes linéaires	70
3.2.3	Généralisation	72
3.2.4	Contraintes supplémentaires non-linéaires	72
3.2.5	Résultats numériques	74
3.2.6	Contrainte sur l'angle de manœuvre de l'avion dévié	74
3.3	Comparaison offset - point tournant	78
3.4	Avions en rattrapage	79
3.5	Résolution en vitesse	81
3.5.1	Etude théorique pour des avions stables en croisière	81
3.5.2	Cas des avions en descente	85
3.6	Modélisation de l'incertitude.	90
3.6.1	Modélisation	90
3.6.2	Evaluation de la probabilité de conflit	90
3.6.3	Espérance de coût de résolution de conflit	92
3.7	Modélisation temps réel	96
3.7.1	Modèle de cluster	97
3.7.2	Déroulement en temps réel	98
4	Résolution numérique par Algorithmes Génétiques	101
4.1	Principes généraux	102
4.2	Résultats théoriques	104
4.3	Description détaillée	105
4.3.1	Codage des données	105
4.3.2	L'algorithme génétique élémentaire	106
4.3.3	Génération de la population initiale : <i>INITIALISER</i> [POP]	106
4.3.4	Evaluer la population : <i>EVALUER</i> [POP]	107
4.3.5	L'opérateur de sélection : <i>SELECTION</i> [NPOP, POP]	107
4.3.6	Opérateur de Croisement : <i>CROISEMENT</i> [NPOP]	109
4.3.7	Opérateur de mutation : <i>MUTATION</i> [NPOP]	112
4.4	Améliorations classiques	113
4.4.1	Introduction	113
4.4.2	Scaling : <i>SCALING</i> [NPOP]	114
4.4.3	Sharing	115
4.4.4	Sharing clusterisé : <i>SHARING</i> (NPOP)	118
4.4.5	Elitisme	119
4.4.6	Algorithmes génétiques et recuit simulé	119
4.4.7	Recherche des optima de Pareto	121
4.4.8	L'algorithme génétique amélioré retenu	123
4.5	Fonctions partiellement séparables : croisement adapté	123
4.5.1	Introduction	123
4.5.2	Problèmes concernés	124
4.5.3	L'opérateur de croisement adapté	124
4.5.4	Application à un polynôme	125

4.5.5	Une fonction trompeuse	128
4.5.6	Application à la fonction de Corana	130
4.5.7	Application à la fonction de Griewank	131
4.6	Application au problème de résolution de conflits	134
4.6.1	Codage des données	134
4.6.2	Gestion des contraintes annexes : $VERIF[POP(i).data]$	134
4.6.3	Initialisation de la population	140
4.6.4	Fitness	140
4.6.5	Opérateurs de croisement et de mutation adaptés	142
4.6.6	Sharing simplifié	144
4.6.7	Méthode locale en fin de convergence	144
4.6.8	Applications numériques	145
4.6.9	Qualité des solutions obtenues	158
4.6.10	Recherche des optima de Pareto	163
A	Rappels mathématiques	169
B	LANCELOT	173
B.1	$SBMIN$	173
B.2	$AUGLG$	176
C	Comparaison point tournant - offset	179
D	Point tournant : étude dans \mathbb{R}^4	183
D.1	Calcul des solutions admissibles et de leur fonction coût	183
D.2	Recherche du coût minimum global	184
D.3	Représentation de l'espace des solutions admissibles	184
E	AG : résultats théoriques	193
E.1	Les algorithmes génétiques à codage binaire	193
E.1.1	Définitions fondamentales	193
E.1.2	Effets de la reproduction	194
E.1.3	Effets des croisements	195
E.1.4	Effets des mutations	195
E.1.5	Conclusions	196
E.2	Modélisation par chaîne de Markov	196
E.2.1	Modélisation de l'algorithme génétique	196
E.2.2	Description rapide de l'algorithme	197
E.2.3	Modélisation	198
E.2.4	La théorie de Freidlin et Wentzell	203
E.2.5	Résultats de convergence	207

Introduction

Le trafic aérien a connu une progression très importante pendant les dernières décennies. Le nombre de données à traiter augmentera encore avec la mise en place de nouveaux moyens de communication entre le sol et les avions (Data-Link). Parallèlement, les avions s'équipent de moyens sophistiqués de navigation (FMS¹, GPS²) permettant des tenues plus précises de trajectoires.

Dans ce cadre, il faut s'interroger sur l'évolution des techniques de contrôle pour les années à venir. On se retrouve alors pris entre deux positions caricaturales : l'automatisation complète ou le refus complet de toute automatisation. Les tenants de l'une ou l'autre approche ont des discours qui relèvent plus de l'émotionnel que du rationnel.

On peut évoquer des arguments en faveur de l'automatisation : l'opérateur humain est actuellement le goulot d'étranglement du système de contrôle. On peut certes espérer améliorer ses capacités de traitement en améliorant la présentation des informations ; mais si l'augmentation du trafic se poursuit dans les mêmes proportions, il faudra déléguer au moins partiellement les tâches de résolution et de surveillance à la machine.

On peut également alimenter les discours dans le sens inverse : une automatisation complète est irréaliste dans le contexte actuel, pour tout un ensemble de raisons : certaines ont un caractère politique ou social : la transition d'un contrôle où l'homme a une responsabilité complète à un contrôle où il n'aurait qu'une activité de supervision pose de sérieux problèmes éthiques. D'autre part, il n'existe pour l'instant aucun système capable de résoudre l'ensemble des problèmes techniques qui se posent dans ce cadre.

La voie moyenne consiste sans doute à penser un système qui, dans un premier temps, serait susceptible de fournir une aide à la détection et à la résolution des conflits, et pourrait, à la demande de l'opérateur, effectuer automatiquement certaines résolutions. Le Centre d'Etudes de la Navigation Aérienne s'est intéressé à cette hypothèse dans le cadre de certains de ses projets.

D'autres projets, comme ARC-2000 (développé par le Centre Expérimental Eurocontrol de Brétigny), ont exploré des voies plus futuristes, tout en souhaitant utiliser dans les systèmes de contrôle actuels, les algorithmes et concepts dévelop-

1. Flight Management System

2. Global Positioning System

pés par des équipes de recherche pour les systèmes de contrôle du futur.

Quelles que soient les voies envisagées et les solutions proposées par les uns et les autres, les problèmes techniques à résoudre restent les mêmes ; il faut savoir prévoir les trajectoires des avions, détecter les conflits, regrouper ces conflits par paquets, et donner des ordres de manœuvre simples les résolvant, etc. Dans cette thèse, on s'intéresse au problème d'optimisation de trajectoires pour la résolution de conflits en route.

Dans la première partie, quelques définitions utiles permettront au lecteur de se familiariser avec le monde de l'ATC³. Nous précisons le contexte du problème, les objectifs visés et les contraintes que nous nous sommes fixées. La partie 2 propose une étude mathématique du problème de résolution de conflits à deux avions dans le plan. Les résultats sont illustrés numériquement. L'efficacité d'une méthode locale d'optimisation non linéaire sous contraintes utilisant le Lagrangien augmenté est testée. Dans la partie 3, les résultats de la partie 2 sont exploités afin de proposer une modélisation du problème répondant aux objectifs et respectant les contraintes fixées dans la partie 1. La partie 4 présente les algorithmes génétiques, justifie leur intérêt pour le problème de résolution de conflits, propose un opérateur de croisement adapté au problème de résolution de conflits et teste sur différentes configurations de conflits les performances des algorithmes génétiques.

Cette thèse a fait l'objet de plusieurs publications [DAAS94, DAN94, MDA94, DAN96b, DCA96, DAN96a, MDA96, DDAS96].

3. Air Traffic Control

Chapitre 1

Le contrôle du trafic aérien

1.1 Définitions

Le contexte ATC du problème met en jeu un certain nombre de termes dont il est utile de préciser le sens :

Contrôle en route : il s'agit du contrôle à l'extérieur des zones entourant les aéroports (dans ces zones, on parle de *contrôle d'approche*). Les avions utilisent alors le plus souvent des routes pré-établies, les *couloirs aériens* (en anglais *airways*). Ces couloirs sont des *tubes* de sections rectangulaires, entourant des segments de droites, aux intersections desquels sont situées des *balises*. C'est souvent autour de ces balises qu'apparaissent des *conflits*. Ces routes contournent les parties d'espace réservées aux militaires et permettent aux contrôleurs d'avoir une visualisation plus aisée de la situation spatiale des avions.

Plan de vol : il contient tous les éléments indicatifs décrivant le vol prévu pour un avion. Ces informations sont déposées avant le départ auprès des services ATC avec, notamment, les éléments suivants :

- l'heure de départ ;
- le niveau de vol¹ demandé pour la croisière ;
- la route prévue : elle est décrite par une série de balises.

Séparations standard : on définit une distance horizontale, qui est exprimée en milles nautiques² (NM), la *séparation horizontale*, et une distance verticale, qui, elle, est exprimée en pieds³ (ft) : la *séparation verticale*. On dit que deux avions sont *séparés* quand la distance qui sépare leurs projections sur un plan horizontal est supérieure à la séparation standard horizontale OU

1. Le niveau de vol ou altitude-pressure est l'altitude de l'avion exprimée en centaines de pieds. Par exemple le FL290 (FL pour Flight Level) correspond à une altitude de 29000 pieds soit environ 9000 m.

2. 1 mille nautique vaut 1852 mètres (1 minute d'arc sur un méridien).

3. 1 pied vaut 30,48 cm.

quand la distance qui sépare leurs projections sur un plan vertical est supérieure à la séparation standard verticale. Ces *séparations standard* peuvent être différentes selon le type de zone dans laquelle se trouvent les avions. Dans le cas du trafic *en route*, qui nous occupe ici, la séparation standard horizontale est de 8 *NM* mais devrait bientôt descendre à 5 *NM*. La séparation standard verticale vaut 1000 ft au dessous du FL 290 et 2000 au dessus. Les vitesses seront exprimées en nœuds (*kt*), à savoir des milles nautiques par heure.

Conflit élémentaire : deux avions sont dits en *conflit* quand ces avions ne sont plus *séparés*, c'est à dire quand les distances séparant leurs projections sur un plan horizontal et sur un plan vertical sont inférieures respectivement à la séparation standard horizontale et à la séparation standard verticale. De façon plus générale, si l'on se fixe une durée T , deux avions seront dits en conflit potentiel pendant T si durant le temps T ils ont une probabilité non nulle d'être en conflit.

Cluster : un *cluster* d'avions est une fermeture d'avions en conflits potentiels. Si l'avion A est en conflit potentiel avec l'avion B et si l'avion B est en conflit potentiel avec les avions C et D , alors on dit que les avions A , B , C et D appartiennent au même cluster. On trouvera par la suite l'expression *conflit à n avions* qui signifie en fait cluster à n avions.

1.2 Contexte du problème

Les organismes du contrôle aérien sont responsables de l'écoulement du trafic dans l'espace dont ils ont la charge. Le service rendu aux usagers doit offrir une sécurité parfaite mais également le meilleur débit possible. Pour cela, l'espace est découpé en *secteurs* dont une position de contrôle a la charge. A l'intérieur de chaque secteur, les contrôleurs maintiennent chaque avion séparé du reste du trafic en donnant aux pilotes des instructions suivies par ces derniers. Le *contrôleur organique* assure la coordination avec les secteurs voisins et fait de la pré-détection de conflits. Le *contrôleur radar* surveille le trafic, prend les mesures d'évitement nécessaires et communique avec les pilotes.

Fluidifier le trafic n'est pas de la seule responsabilité des contrôleurs. Le contrôle en route se situe au sein d'une chaîne de filtres successifs où chaque élément essaie d'améliorer l'écoulement du trafic. Chaque filtre a des objectifs différents et gère des espaces et des horizons temporels distincts. On peut grossièrement distinguer cinq niveaux :

Organisation à long terme : c'est le filtre le plus grossier. Son but n'est pas au sens strict d'éviter des conflits mais plutôt d'organiser le trafic de façon macroscopique à moyen et long terme (supérieur à 6 mois). On peut citer à titre d'exemple, les schémas d'orientation de trafic, les mesures du Comité des horaires ou encore, les accords inter-centres et les accords avec les militaires qui permettent aux civils d'utiliser leurs zones aériennes pour écouler les pointes de trafic du vendredi après-midi.

Organisation à court terme : on parle souvent de pré-régulation. Elle consiste à organiser une journée de trafic j , la veille ($j - 1$) ou l'avant-veille ($j - 2$). On dispose pour cela de données relativement précises :

- les plans de vol déjà connus ;
- la capacité de contrôle que peut offrir chaque centre en fonction des effectifs qui seront présents le jour j ;
- le débit maximal d'avions pouvant pénétrer dans un secteur pendant une période de temps donnée, encore appelé capacité du secteur ;
- les données des années et des semaines précédentes. En effet, le trafic aérien est relativement répétitif : le lundi ressemble au lundi de la semaine précédente, le pont de l'Ascension ressemble à celui de l'année passée, etc. Ceci permet de prévoir où vont apparaître les congestions, la capacité qu'il va falloir mettre en œuvre pour répondre à la demande, voire éventuellement les mesures plus contraignantes à prendre.

Ce filtre a non seulement une action macroscopique car il organise les flux de trafic en fonction des capacités offertes mais également une action ponctuelle sur chaque avion en gérant les créneaux de décollage⁴. Effectué au niveau

4. Un créneau de décollage est une fenêtre temporelle pendant laquelle l'avion est autorisé à décoller.

national jusqu'en 1995, ce filtrage est maintenant fait au niveau européen pour obtenir une meilleure coordination. C'est le rôle de la CFMU⁵.

Régulation en temps réel : ce filtre consiste à organiser les différents flux en tenant compte des événements du jour. Il s'agit plutôt de mesures d'ajustement qui prennent en compte des événements encore mal connus la veille. Ainsi, le trafic transatlantique est mal connu à $j - 1$ mais est bien connu entre 3 à 6 heures avant son arrivée. La fraction de la capacité offerte qui avait été réservée par la pré-régulation peut alors être adaptée. On peut envoyer sur d'autres centres les avions supplémentaires ou augmenter le nombre de vols non transatlantiques traités par le centre s'il y a moins de vols transatlantiques que prévus. De la même façon, on peut ré-allouer des créneaux horaires non utilisés pour une raison quelconque (retard ou incident technique) ou tenir compte de la météo du jour (terrains inaccessibles par exemple). Ce rôle est en général joué par les FMP⁶ dans chaque centre.

Tactique : c'est le dernier filtre de la chaîne du contrôle aérien : l'action du contrôleur sur son secteur. Le temps moyen passé par un avion dans un secteur est de l'ordre d'une quinzaine de minutes. La visibilité du contrôleur est un peu supérieure puisqu'il dispose des plans de vol quelques minutes avant l'entrée de l'avion dans le secteur.

La détection, et surtout la résolution des conflits, ne sont pas automatisées. Les contrôleurs sont donc entraînés à reconnaître des types de conflits et à appliquer à ces derniers des manœuvres connues. Le contrôleur ne peut éviter les conflits qu'en jouant sur la trajectoire des avions. Pour cela, il dispose de quatre types d'ordres :

- des déviations de cap ;
- des paliers intermédiaires pour les avions en montée ou en descente ;
- des changements de niveaux de vol.
- des mesures de régulation de vitesse (essentiellement sur des avions en descente) ;

Cependant les résolutions proposées ne sont pas toujours optimales. En outre, une fois une manœuvre d'évitement choisie, le contrôleur doit également surveiller les avions impliqués pour vérifier que tout se passe bien. Tout ceci se fait au détriment de la charge⁷ de travail du contrôleur et affecte donc la capacité du secteur dont il est responsable.

Urgence : Ce filtre n'est censé intervenir que lorsque le système de contrôle est absent ou a été défaillant.

- Le *filet de sauvegarde* prédit la trajectoire de chaque avion avec un horizon temporel de quelques minutes à l'aide des positions radar passées et

5. Central Flow Management Unit

6. Flow Management Position

7. La charge de travail prend en compte des tâches de surveillance, communication, détection et résolution des conflits.

d'algorithmes de poursuite. En cas de conflit, il déclenche une alarme. Il ne propose pas de solution aux conflits détectés.

- TCAS⁸ : l'objet de ce filtre n'est plus de séparer les avions mais plutôt d'éviter une collision présumée. La prédiction temporelle est inférieure à la minute et varie entre 25 et 40 secondes. Il est trop tard pour que le contrôleur intervienne puisque l'on estime qu'il lui faut entre 1 et 2 minutes pour analyser une situation, trouver une solution et la communiquer aux avions. Actuellement, le TCAS détecte les avions environnants et donne un ordre de résolution au pilote (pour le moment dans le plan vertical). Ce filtre doit résoudre les conflits non prévisibles comme par exemple, un avion dépassant un niveau de vol donné par le contrôle, ou un accident technique qui dégraderait notablement les performances de l'avion.

8. Traffic alert and Collision Avoidance System : système embarqué à bord de certains avions que les Etats-Unis ont rendu obligatoire pour tous les avions de plus de 30 passagers.

1.3 Les projets d'automatisation existants

1.3.1 La tentative américaine AERA

Les travaux de recherche de la MITRE sur AERA⁹ [Cel90, SPSS83, NFC⁺83, Nie89b, Nie89a, PA91] sont financés par la FAA¹⁰. La première phase de ce projet, AERA 1 permettait de prévoir les trajectoires des avions en fonction des intentions des pilotes et de détecter d'éventuelles violations de la séparation standard ou de restriction de flux pour des plans de vol existants ou désignés par un contrôleur. C'est donc avant tout un outil d'aide à la prise de décision qui ne propose pas de solution aux opérateurs.

AERA 2 [Cel90] propose d'importantes nouvelles aides aux opérateurs, et notamment une liste de "résolutions recommandées par l'ordinateur" aux éventuels conflits élémentaires détectés par AERA 1. AERA 2 introduit également des outils d'aide à la coordination entre contrôleurs¹¹. Néanmoins, jusqu'à ce jour, dans AERA 2, la responsabilité de séparer les avions relève en dernier ressort du contrôleur.

Dans AERA 3 [NFC⁺83, Nie89b, Nie89a, PA91], la responsabilité de séparer les avions est laissée à la machine. La structure d'AERA 3 est hiérarchique. Au niveau national, l'ATMS¹² reste en l'état. Il assure une gestion des flux de trafic acceptable pour AERA 3.

AERA 3 se décompose en trois niveaux hiérarchiques :

- Le niveau supérieur, juste en dessous de l'ATMS est l'AMPF¹³. Son rôle est de fournir des outils d'aide à la décision à un opérateur humain appelé "Airspace Manager". Ses fonctions sont semblables à celles de l'actuelle gestion locale des flux. L'AMPF a une gestion macroscopique et statistique du trafic. La plage d'anticipation est de l'ordre de 90 minutes, temps suffisamment long pour que les incertitudes sur les trajectoires d'avions ne permettent pas de prévoir les conflits individuels. Parmi les fonctions d'AMPF, "Density/Complexity Manager" interagit avec le niveau inférieur, le MOM¹⁴. Il détermine les régions où le trafic est trop dense ou trop complexe pour le MOM et propose des manœuvres pour résoudre ce problème.
- Le niveau intermédiaire est le "Maneuver Option Manager". MOM a un temps d'anticipation compris entre 5 et 30 minutes. Les conflits peuvent alors être détectés et analysés efficacement dans leur contexte. En fonction des incertitudes, MOM permet de déterminer quelles paires d'avions posent des problèmes et quelles sont les meilleures manœuvres à effectuer pour un problème spécifique.

9. Automated En-Route Air Traffic Control

10. Federal Aviation Administration

11. Les contrôleurs surveillent et interviennent sur des secteurs de contrôle, quand un avion passe d'un secteur à l'autre, il doit être libéré par le premier et être pris en compte par le suivant.

12. Air Traffic Management System

13. Airspace Manager Planning Functions

14. Maneuver Option Manager

AERA 3 reprend les notions de conflits potentiels ou “Possibles”¹⁵ et de clusters, fermeture transitive de possibles. AERA 3 propose des AMO¹⁶. Pour cela, pour chaque avion, il détermine quelles sont les manœuvres “libres de possibles”. Ces manœuvres sont limitées à certains déplacements : à gauche, à droite, au dessus, en dessous de la trajectoire nominale, plus vite ou moins vite. Une trajectoire admissible pour un avion doit éliminer tous les possibles de cet avion sans en créer de nouveau. AERA 3 suppose que les avions respectent leurs trajectoires avec une certaine tolérance.

MOM simplifie les clusters en garantissant l'existence d'AMO pour un ou plusieurs avions impliqués dans un cluster. L'objectif est de pouvoir réduire le cluster en plusieurs clusters indépendants, plus petits et moins complexes à résoudre.

MOM définit un “plan de contingence”, à savoir un ensemble d'AMO qui permettent de séparer toutes les paires d'avions. Le plan de contingence simplifie tous les clusters à un niveau suffisant pour permettre au niveau inférieur d'AERA 3 de résoudre tout possiblem restant. Il s'agit donc d'isoler des possibles admettant des solutions. On peut également conserver des clusters plus compliqués dans la mesure où ils contiennent des conflits plus éloignés dans le temps.

Occasionnellement, MOM identifie ou désigne une AMO particulière qui apporte suffisamment d'amélioration au système pour pouvoir être décidée à l'avance. Un des problèmes à résoudre est de déterminer quelle AMO désigner. Les bons candidats peuvent par exemple être des AMO qui résolvent beaucoup de possibles qui ont une forte probabilité de devenir des conflits réels. Les réponses à ces problèmes restent très floues dans les rapports de la MITRE.

- Le niveau inférieur d'AERA 3 est l'ASF¹⁷. Le rôle d'ASF est de déterminer des résolutions détaillées pour des problèmes entre **deux** avions ou entre un avion et un espace interdit. Il cherche une solution sûre qui minimise la pénalisation en temps ou en carburant. Il gère des conflits à une échéance inférieure à 10 minutes, ce qui réduit notablement les problèmes liés à l'incertitude sur la position des avions. ASF gère les possibles un par un et indépendamment. ASF considère en entrée les données fournies par MOM et respecte les AMO désignées par MOM.

En résumé :

- ASF sépare les paires d'avions.
- MOM assure le respect du contexte global par ASF (bien qu'ASF ne gère que des paires d'avions séparément).
- L'AMPF s'assure que MOM peut opérer avec succès. En prévenant les trop fortes densités de trafic, il permet d'éviter que les avions aient trop peu

15. Par possiblem il faut comprendre problèmes possibles en raison des incertitudes liées aux trajectoires de chaque avion, mais qui ont une probabilité non nulle de se résoudre sans manœuvre

16. Available Maneuver Options

17. Automated Separation Function

d'AMO.

- L'ATMS, au niveau national, assure que l'AMPF peut gérer le trafic.

ASF et MOM sont les deux niveaux dont l'automatisation complète est envisagée. Néanmoins, le seul niveau détaillé clairement d'AERA 3 est l'ASF qui ne résout que des conflits à deux avions. Les objectifs de MOM sont assez clairement définis. Par contre, son fonctionnement reste très flou et peu convaincant. La recherche d'optimalité globale n'apparaît pas du tout dans AERA 3.

L'ASF est assurée par un algorithme appelé "Gentle-Strict" (GS) dont le but est de résoudre automatiquement des conflits de croisement horizontal entre deux avions en utilisant des manœuvres latérales d'offset¹⁸. GS prévoit :

- une manœuvre d'offset latéral pour un des deux avions dit "gentle" selon un sens prédéterminé.
- une consigne de tenue de trajectoire pour l'autre avion dit "strict" dans le but d'éviter les déviations supérieures à un certain seuil de tolérance par rapport à sa trajectoire nominale.

GS ne résout un conflit que dans le plan horizontal et ne change jamais les vitesses des avions ou leurs niveaux de vol. GS agit le plus tard possible afin d'éviter les manœuvres inutiles.

Les paramètres pris en compte par GS pour la résolution d'un conflit à deux avions sont les suivants :

- θ , l'angle entre les deux trajectoires.
- v , la vitesse de l'avion le plus rapide.
- r , le rapport de vitesse entre l'avion le plus lent et l'avion le plus rapide.
- T , le temps qui sépare le passage des deux avions à l'intersection des trajectoires.
- $Latl, Latr, Lonl, Lonr$, les bornes de l'incertitude sur les positions latérales et longitudinales des deux avions (l pour "lent" et r pour "rapide") sur les 10 minutes de prédiction.
- la "gentleness", que l'on pourrait appeler "pénalisation" de la manœuvre de résolution paramétrée par la valeur de l'offset u et par le retard induit par la mise en offset v .
- "NEED - SEP", la norme de séparation requise.
- "GS - SEP", la séparation obtenue.

GS distingue 4 situations différentes : l'avion le plus lent à l'extérieur, l'avion le plus lent à l'intérieur, l'avion le plus rapide à l'extérieur, l'avion le plus rapide à l'intérieur. GS a deux objectifs : trouver la manœuvre optimale pour un conflit et une NEED - SEP donnée et calculer sa pénalisation ou trouver la séparation réalisable pour un conflit avec une pénalisation maximale donnée. Résoudre ces deux problèmes revient à résoudre un problème linéaire simple qui peut se résumer à une demi-page de calculs. Les résultats sont très bons dans la plupart des cas, ils deviennent mauvais quand $\theta \in [0, \pi/4]$ et $r \in [0.85, 1]$.

Il semble aujourd'hui que le projet AERA ait été arrêté. Les algorithmes utilisés

18. un offset est une mise en parallèle par rapport à la trajectoire initiale de l'avion

ou envisagés dans le MOM n'ont cependant jamais été évoqués dans les rapports disponibles. Sur le plan de la modélisation, AERA3 peut garder un intérêt. Par contre, ce projet n'apporte pas de réponse au problème d'optimisation globale dès que le nombre d'avions dépasse 3.

1.3.2 ARC2000 et ses dérivés

Le projet d'automatisation complète du contrôle en route européen, ARC2000 [K⁺89, FMT93] a connu de profonds changements depuis son origine. Néanmoins la modélisation en "tubes 4D" des trajectoires d'avions est restée et a permis de définir des outils d'aide à la décision très innovants et efficaces pour le contrôleur [MG94].

Si l'on est capable de prédire avec une très bonne précision la trajectoire d'un avion, on peut représenter celle-ci par une courbe dans \mathcal{R}^4 , les trois premières variables représentant la position dans l'espace de l'avion au temps t , quatrième variable. Les imprécisions de mesures de positions et de tenues de trajectoires 4D transforment, dans un souci de réalisme, cette courbe en "tube" en lui donnant une certaine section. Pour séparer les trajectoires d'avions, il faut donc fabriquer un "tube" pour chaque avion de sorte que les différents tubes soient d'intersection vide. Pour cela, ARC2000 utilise un algorithme d'optimisation locale (type "gradient") qui permet, étant donné n tubes d'intersection vide, de construire un tube $n + 1$ d'intersection vide avec tous les précédents et minimisant le retard de l'avion $n + 1$.

Le principe d'ARC2000 était donc originellement le suivant : le premier avion entrant dans le système se voyait affecter un tube optimal respectant son plan de vol. Dès qu'un nouvel avion se présentait, son tube optimal était calculé en considérant les tubes précédemment affectés comme des contraintes fixes. Autrement dit, un tube déjà affecté n'était pas remis en cause. L'optimalité globale n'était donc pas recherchée. Dans le cas où un avion ne respectait pas le tube qui lui était affecté, il devait négocier un nouveau tube respectant tous les autres tubes. La faiblesse de ce principe se caractérisait par son manque de robustesse. En effet, il est possible que le non respect d'un tube 4D soit lié à des événements météorologiques par exemple, et qu'il ne touche pas seulement un avion mais plusieurs. On pouvait alors craindre un phénomène chaotique qui remettait en cause le principe adopté.

Les hypothèses de départ d'ARC2000 le projetaient bien au-delà de l'an 2000. En effet, la négociation des tubes 4D suppose que les avions soient équipés de FMS-4D¹⁹, ce qui n'est pas réaliste à court terme.

Le plan stratégique d'ARC2000 consiste à garantir des trajectoires sans conflit pour les 20 minutes à venir. Il faut en effet trouver un juste milieu entre prévoir des trajectoires sans conflit très longtemps à l'avance, et ne pas tenir compte des conflits à l'avance et réajuster continuellement les trajectoires. Le compromis d'ARC2000 consiste à prévoir la trajectoire dans sa totalité avec le moins possible de conflits tout en surveillant les conflits potentiels. 20 ou 30 minutes à l'avance,

19. Flight Management System en 4 dimensions

les conflits sont éliminés. Avant cette période, seuls les conflits qui sont fermement diagnostiqués peuvent entraîner une réorganisation des trajectoires.

N'ayant pas de rapport écrit récent sur l'évolution du projet ARC2000, les éléments qui vont suivre sont susceptibles d'être réactualisés.

Il semble que l'évolution d'ARC2000 se tourne vers des hypothèses plus réalistes. La modélisation de la trajectoire totale sans conflit (de l'origine à la destination) semble avoir été abandonnée pour une gestion à 20 ou 30 minutes des groupes de conflits ou clusters. La recherche de solutions optimales a également évolué puisqu'au principe du "dernier arrivé dernier servi", a succédé une série de règles qui permettent de classer les avions du plus prioritaire au moins prioritaire et les manœuvres de la plus opportune à la moins opportune. On retiendra essentiellement que la recherche d'un optimum global est toujours délaissée au profit d'un parcours d'arbre dans un ordre régi par des règles empiriques.

Il est à noter que les recherches d'Eurocontrol ont permis de produire des outils d'aide à la décision très innovants tels qu'HIPS²⁰ [MG94], qui ne s'intéresse plus à l'optimisation automatique de trajectoire mais donne au contrôleur des outils d'aide à la représentation des trajectoires et des conflits.

On notera enfin qu'ARC2000 est le seul projet ayant été testé sur le trafic réel. Comme pour AERA3, la modélisation du problème est intéressante mais les algorithmes de résolution de clusters ne s'attaquent pas au problème d'optimisation global des trajectoires.

1.3.3 Le projet SAINTEX

Dans le projet SAINTEX²¹ [AL92], trois approches d'automatisation du contrôle en route sont abordées.

- Le scénario "Détection-Résolution" est un système orienté système expert. Le système essaie de reproduire le comportement du contrôleur. Les conflits sont détectés en extrapolant les trajectoires d'avions 10 minutes dans le futur (6 minutes pour les avions évolutifs²²). Aussitôt détecté, un conflit est classé suivant différents critères tels que l'angle formé par les trajectoires, le rapport des vitesses, etc. Pour chaque classe de conflits, une manœuvre prédéfinie est appliquée. Le système expert ne peut résoudre qu'un conflit entre deux avions.
- Dans le scénario "4D", une trajectoire sans conflit est générée pour tout avion entrant dans le secteur. Une trajectoire est représentée par un ensemble de points et de contraintes verticales. Pour chaque avion, on construit un tube 4D qui représente sa trajectoire compte tenu des incertitudes sur sa vitesse et sa position. Pour construire un tube admissible (sans conflit), SAINTEX fait diverses tentatives partant de la trajectoire idéale (directe) vers des trajec-

20. Highly Interactive Problem Solver

21. SAINTEX fait référence au vol de nuit. Le projet était prévu pour être mis en œuvre avec de faibles densités de trafics, comme la nuit, par exemple.

22. Un avion évolutif, contrairement à un avion dit "stable" est soit en montée, soit en descente.

toires de plus en plus pénalisantes mais résolvant les conflits. Du temps que l'on consacre à la recherche d'une trajectoire admissible découle la qualité de la résolution. La trajectoire de l'avion est ensuite surveillée de manière à s'assurer que l'avion respecte bien la trajectoire qui lui a été affectée. Ce scénario, purement algorithmique, ressemble fortement à celui décrit dans ARC2000.

- Dans le scénario hybride, les avions stables sont gérés par le système Détection-Résolution et les autres par le système 4D.

Utiliser un système expert pour résoudre un conflit à deux avions ne se justifie pas. Le projet SAINTEX, par définition, s'intéresse à la gestion automatique d'espaces aériens peu saturés. Le problème de la résolution des clusters d'avions est évoqué, mais non résolu.

1.3.4 Les techniques réactives

Les méthodes de résolution décrites dans ce paragraphe sont radicalement différentes de celles évoquées précédemment. Le projet ATLAS²³ est parmi les premiers projets à avoir envisagé l'hypothèse d'avions autonomes ou hybrides [DA93]. Cette hypothèse a été étudiée plus en détail par Karim Zeghal [Zeg93, Zeg94] dans sa thèse dans laquelle il introduit la notion de coordination d'actions grâce à différentes forces qui s'exercent sur les agents, dans notre cas, les avions. Il définit ainsi trois types de forces qui devront agir suivant l'urgence :

- Les forces attractives qui permettent aux avions d'atteindre leur objectif (une balise ou leur destination finale par exemple).
- Les forces répulsives qui permettent aux avions d'éviter un obstacle proche donc dangereux. Cet obstacle peut être un avion ou une zone interdite.
- Les forces de glissement qui permettent de contourner les obstacles. Les avions ont alors une action coordonnée (voir figure 1.1). Une force de glissement est définie de la manière suivante : si l'on observe l'équipotentielle de danger passant par l'avion, une force de glissement est tangente à cette équipotentielle alors que la force répulsive est normale à cette équipotentielle. Il y a donc plusieurs forces de glissement possibles. Si l'on reste dans le plan horizontal, la figure 1.1 montre que l'on peut définir deux sens de glissement (à droite ou à gauche). Le sens optimal (il s'agit ici d'optimalité locale, pour l'avion concerné) est celui qui favorise le rapprochement vers l'objectif. Dans le cas d'un obstacle mobile, Karim Zeghal définit une action coordonnée d'évitement. Si deux avions arrivant au même point suivent des forces de glissement complémentaires, alors l'évitement peut se faire de façon très efficace. Dans le cas simple de deux avions (voir figure 1.1), on projette par exemple la vitesse relative sur l'équipotentielle de danger pour chaque avion et on obtient ainsi deux forces de glissement coordonnées.

23. Air Traffic Land and Airborne Systems, étude de la CCE/DG XIII, étude de définition d'un système ATM/CNS unique en Europe

FIGURE 1.1 – Force répulsive et force glissante, forces de glissement coordonnées.

Il s'agit ensuite, pour limiter les changements brutaux de direction, de gérer l'intensité des différentes forces. En effet, les avions étant limités par leurs taux de virage, de montée et de descente, ils ne peuvent pas effectuer de manœuvre trop rapide. Les différentes forces s'exerçant sur les avions s'additionnent donc avec des coefficients variables suivant l'imminence du danger.

Pour gérer les conflits à plus de deux avions, Karim Zeghal propose d'additionner les forces relatives à chaque avion. "Intuitivement, excepté quelques cas exotiques, cela devrait permettre d'obtenir des directions de forces cohérentes entre elles et par rapport aux obstacles". Les fondements de cette hypothèse ne reposent que sur l'étude de cas pratiques. Néanmoins, ces résultats pratiques semblent prometteurs pour des faibles densités.

La coordination d'actions grâce aux forces de glissement n'a pas pour objectif la recherche d'optimalité, mais vise, avant tout, une bonne efficacité ainsi qu'une bonne robustesse des solutions.

L'étude des possibilités d'utilisation d'une théorie de la coordination d'actions pour les besoins de la navigation aérienne a commencé au CENA en 1994. Trois systèmes sont envisagés :

- Dans la mesure où le processus individuel ne nécessite que des informations locales à l'appareil, il peut fonctionner de façon autonome à partir d'une perception de son environnement. Karim Zeghal propose donc un système d'avion autonome "hybride"²⁴. L'intérêt premier de ce système est sa robustesse. La panne d'un avion ne remet pas en cause la sécurité du système. Il apparaît une forme de redondance dans le système qui renforce sa robustesse. Ce système devrait gérer des conflits entre 4 et 10 minutes à l'avenir, ce qui suppose qu'un système supérieur continue à gérer la densité locale de trafic afin d'éviter une saturation.

24. L'équivalent du TCAS américain (Traffic alert and Collision Avoidance System) mais à plus longue échéance

- Karim Zeghal propose également un système d'aide au contrôleur. Il suffit pour cela d'utiliser le processus individuel sous forme de processus centralisé et de simuler les trajectoires futures. Le résultat d'une simulation est alors un ensemble de trajectoires assurant l'évitement sur l'intervalle de temps considéré pour l'ensemble des appareils. Ceci pourrait, selon lui, avoir un grand intérêt pour planifier les trajectoires et faire des propositions au contrôleur.

Les résolutions proposées par ces techniques étant des commandes continues, il semble peu probable que celles-ci puissent être mises en pratique dans le cas où le pilote et les contrôleurs sont maintenus dans le système. Il me semble que ce dernier point soit un obstacle majeur à l'utilisation de ces techniques.

- Enfin, on peut imaginer un système double "intermédiaire". Dans ce cas, certains avions sont autonomes et d'autres pas. Le contrôleur a alors une disponibilité supplémentaire et la capacité de son secteur augmente.

La robustesse des solutions est un des atouts des travaux de Karim Zeghal. Il reste néanmoins de nombreux problèmes à résoudre. Dans la mesure où l'on souhaite continuer à confier à l'homme son rôle de pilote dans l'avion, il est indispensable de pouvoir lui transmettre des manœuvres exécutables et donc simples. Par exemple, on peut demander à un pilote de changer de cap pendant un certain temps, de prolonger une montée ou de modifier une heure de début de descente. On ne peut pas lui demander de modifier en permanence son cap, sa vitesse et son taux de montée ou de descente. Il serait donc nécessaire de revoir la modélisation des trajectoires pour les simplifier et les rendre accessibles aux pilotes. De plus, si l'on peut espérer prouver que ces techniques peuvent résoudre tous les conflits pour deux avions, la généralisation à n avions (qui consiste à additionner les forces générées par chaque avion) n'a été vérifiée que sur des exemples. Enfin, aucune recherche d'optimalité globale n'est envisagée.

1.3.5 Conclusions

Il est à noter que les hypothèses de travail des divers projets sont très différentes. Concernant l'équipement des avions, SAINTEX et AERA3 ne font pas d'hypothèse particulière et autorisent des tenues de trajectoires avec une précision compatible avec les performances actuelles. Néanmoins, ces deux projets semblent avoir été plus ou moins abandonnés. ARC2000 semble très prometteur, même si l'équipement des avions qu'il requiert le place dans un futur assez éloigné. Le projet des avions réactifs de Karim Zeghal, apparemment prometteur, nécessitera des adaptations conséquentes (simplification des trajectoires) pour laisser au pilote la maîtrise de l'avion.

La recherche d'optimalité globale n'est prioritaire dans aucun des projets, AERA3 est hiérarchisé et ne cherche une solution optimale que dans le cas d'un conflit entre deux avions. ARC2000 parcourt l'espace des solutions en s'appuyant sur une table de priorités. Les manœuvres sont classées par ordre de priorité et testées dans cet

ordre-là. Il semble d'ailleurs que cette recherche ne permette pas, même pour des densités raisonnables de trafic, de résoudre 100% des conflits. Les projets "systèmes experts" du type SAINTEX se rapprochent des solutions optimales empiriquement connues pour des conflits à deux avions mais ne s'intéressent pas à une recherche d'optimalité. Les avions réactifs de Karim Zeghal donnent de "bonnes solutions" mais ne recherchent pas de solution optimale.

Les délais temporels d'anticipation des trajectoires sont souvent semblables (entre 20 et 30 minutes), à l'exception du projet de Karim Zeghal plus réactif avec une anticipation entre 5 et 10 minutes, mais ne sont pas toujours justifiés rigoureusement. Il est étonnant qu'aucun des rapports précédemment cités ne se soit intéressé à relier quantitativement l'intérêt de prévoir un conflit longtemps à l'avance avec des paramètres tels que la précision des avions en tenue de vitesse, de cap, d'altitude. EUROCONTROL, dans un rapport plus récent [AS96] s'y intéresse. Nous essaierons d'apporter également quelques éléments de réponse dans la partie 3.

1.4 Hypothèses de travail

L'objet de ces travaux est de résoudre de façon globalement optimale des conflits aériens ou clusters d'avions, en vue de créer un outil d'aide à la décision ou d'automatisation. La modélisation abordée chapitre 3 devra tenir compte des hypothèses de travail détaillées dans cette partie.

Il a été décidé de travailler dans le cadre du filtre tactique, du moins dans un premier temps (certains éléments du système pourraient cependant être communs avec un système élaboré pour le filtre pré-tactique). Il faut préciser la notion d'optimalité d'une manœuvre de résolution. Plusieurs types de critères peuvent être appliqués pour juger une déviation.

Des critères objectifs :

- Minimiser la distance supplémentaire parcourue par l'ensemble des avions détournés²⁵.
- Minimiser la durée de l'ensemble des déviations.
- Minimiser le coût global des déviations, en tenant compte des consommations et des retards occasionnés.
- Minimiser le nombre de manœuvres à effectuer.

Des critères subjectifs :

- Vaut-il mieux dévier un seul avion de façon importante que deux ou plusieurs avions d'une quantité plus faible ?
- Vaut-il mieux dévier faiblement un avion avec 400 passagers que fortement un avion avec 50 passagers ?
- Vaut-il mieux dévier un long-courrier qu'un court-courrier dont le vol est plus économique mais qui est plus sensible aux retards ?

Ces listes ne sont pas exhaustives. Lorsqu'un contrôleur prend une décision, il applique des règles objectives et subjectives pour prendre la meilleure décision. Certaines de ces règles sont bien connues, d'autres sont plus difficiles à verbaliser et font appel à l'expérience du contrôleur. Il n'existe donc pas de solution optimale unique pour tous les conflits.

1.4.1 Premières spécifications

Ayant défini notre objectif, nous pouvons dégager les premières spécifications du système :

- Le système est destiné à être intégré dans le service rendu par le contrôle aérien. C'est donc un système *sol* qui utilisera les données actuellement connues par le système sol : position radar des avions et plans de vol.
- Le système ne fera pas d'hypothèse sur l'équipement des avions et notamment sur leur capacité de navigation 4D ou de systèmes de communication bord-sol.

25. Pour donner un ordre de grandeur, on pourra considérer qu'une heure d'A320 coûte environ 50000 francs, ce qui ramène la minute à plus de 800 francs et le nautique à 100 francs.

- On désire pouvoir travailler dans un environnement *temps réel*. Cette spécification va avoir trois grandes conséquences sur notre système :
 - Le système va devoir être *prédictif*. En effet, la détection et la manœuvre d'évitement nécessaire doivent s'effectuer avant le conflit réel. Le système connaîtra l'ensemble des positions et des plans de vol à un instant t donné et devra en déduire les trajectoires que les avions seront supposés suivre entre t et $t + d$. La durée d sera appelée *fenêtre d'anticipation* et sera notée par la suite F_a .
 - Le programme chargé de résoudre les conflits sera *itératif* et devra pouvoir s'exécuter en un délai de temps Δ fixé. Plus Δ sera grand et plus le programme disposera de temps pour trouver une solution optimale. Plus Δ sera petit et plus la cadence de rafraîchissement des données sera rapide, ce qui permettra de réduire les incertitudes. Δ sera le résultat d'un compromis entre la *vitesse de calcul* désirée et la *profondeur de la recherche* d'une solution optimale.
 - Il faudra prendre en compte le délai nécessaire pour transmettre la manœuvre retenue à chaque avion ainsi que le temps pour que le pilote l'assimile et l'exécute. Ce délai sera compris dans Δ .
- Le système doit optimiser une situation globale et non un simple conflit entre deux avions. Dans la réalité, un contrôleur tient compte du trafic environnant dans les manœuvres d'évitement qu'il donne. Il ne faut pas envoyer un avion a vers un avion c pour éviter b .
- Le système doit pouvoir être compatible avec l'évolution du trafic. Il devra notamment être capable de fonctionner avec des densités de trafic plus importantes que celles actuellement observées.

1.4.2 Contraintes opérationnelles

Les manœuvres d'évitement doivent respecter des contraintes opérationnelles :

- Elles doivent être compatibles avec les performances des avions. Les taux de montée et de descente ainsi que les taux de virages sont bornés.
- Elles ne peuvent pas être continues. De telles manœuvres supposent des ajustements incessants pendant la durée de la manœuvre. Ceci impose l'existence de canaux de données sol-bord bien plus performant que ceux prévus pour les années à venir. En outre, le pilote est "maître à bord" et doit donc valider toute manœuvre demandée par le contrôle. Une telle validation ne serait plus possible avec des manœuvres continues et on se trouverait dans une situation où l'avion serait piloté à partir du sol. Les manœuvres seront donc discrètes et une résolution sera donc une succession limitée de manœuvres.
- Pour des raisons analogues, les manœuvres proposées doivent être simples et standard. Ceci facilitera, d'une part la communication entre le sol et le bord puisqu'il y aura peu d'informations à transmettre, et d'autre part facilitera la tâche de l'équipage qui exécutera toujours les mêmes manœuvres.
- Elles doivent ressembler le plus possible à des manœuvres actuellement em-

ployées par le contrôle.

- Elles doivent être assez robustes pour tenir compte des incertitudes liées à l'utilisation d'un système prédictif ainsi qu'aux erreurs entachant les données initiales. Le taux de fausses alarmes doit être réduit au minimum.
- la fréquence de rafraîchissement des données doit être compatible avec le système de contrôle actuel.
- Les manœuvres d'évitement doivent être surveillées pour éventuellement pouvoir être corrigées en cas d'exécution tardive, incomplète ou incorrecte.

Chapitre 2

Modélisation par Contrôle Optimal

Introduction

L'aéronautique utilise depuis longtemps maintenant les outils du Contrôle Optimal pour optimiser les trajectoires d'un avion [Cle90, MZ88, BH91, ME88, Sch90]. Le Principe du Maximum de Pontriaguine est alors largement utilisé. Le problème de résolution de conflit fait néanmoins intervenir la contrainte de séparation qui exige une forme plus élaborée du Principe du Maximum. Les principaux résultats théoriques sont contenus dans les conditions nécessaires pour les problèmes de contrôle optimal avec contraintes sur l'état (voir annexe A).

Dans la première partie, en s'affranchissant des contraintes sur la vitesse des avions, on montre que le problème de Contrôle Optimal permet de déterminer analytiquement les trajectoires optimales pour un problème à deux avions, et de dégager deux invariants pour le problème à n avions ($n > 2$). Une interprétation géométrique du résultat permet de compléter l'analyse. L'introduction des contraintes sur les vitesses ne permet plus de déterminer la solution analytique du conflit à deux avions. Dans la partie deux, en privilégiant la trajectoire d'un des deux avions (trajectoire rectiligne uniforme), on montre que l'on peut dégager une équation implicite qui donne l'allongement minimal de la trajectoire du deuxième avion. La structure de l'ensemble des solutions admissibles est alors mise en évidence. La partie trois souligne les limites du modèle de contrôle optimal dans le cas de n avions ($n > 2$). La partie quatre donne des résultats numériques effectués sur des conflits théoriques de 2 et 5 avions permettant de confirmer numériquement les difficultés théoriques rencontrées.

2.1 Résolution de conflits : problème de Contrôle Optimal

2.1.1 Conflit entre deux avions non contraints en vitesse

Position du problème

On considère deux avions x et y se déplaçant à altitude constante. Les avions x et y sont repérés par leurs vecteurs position $x(t)$ et $y(t)$ à l'instant t . A $t = t_0$ les deux avions sont en \vec{x}_0 et \vec{y}_0 . A $t = t_f$ les deux avions sont en \vec{x}_f et \vec{y}_f de sorte qu'ils ne prennent pas de retard, quelles que soient les déviations qu'ils auront subies (ces hypothèses fortes vont permettre de donner une résolution analytique du conflit, elles seront bien évidemment abandonnées par la suite). Pour simplifier le problème, on ne fera pas d'hypothèse sur les vitesses initiales et finales des avions. On pose $\dot{x} = \vec{u}$ et $\dot{y} = \vec{v}$ et on cherche la trajectoire optimale en terme de minimisation de la fonction coût $J = \int_{t_0}^{t_f} \frac{u^2 + v^2}{2} dt$ sous la contrainte de séparation des deux avions, à savoir : $\|\vec{x} - \vec{y}\|^2 \geq d^2$. On peut en effet, en première approximation, estimer que la consommation de l'avion est proportionnelle au carré de sa vitesse, d'où la fonction à minimiser choisie.

Le problème de Contrôle Optimal associé

Pour résoudre ce problème analytiquement, on peut l'aborder comme un problème de commande optimale, u et v jouant le rôle de commande. Nous laisserons de côté les flèches sur les vecteurs lorsqu'il n'y aura pas de confusion possible, les indices 1 et 2 indiqueront les composantes sur les deux axes de ces vecteurs. Le problème s'écrit donc :

$$\begin{aligned} \dot{x}(t) &= u(t), \dot{y}(t) = v(t), \forall t \in [t_0, t_f] \\ u(t) &\in \mathbb{R}^2, v(t) \in \mathbb{R}^2, \forall t \in [t_0, t_f] \\ x(t_0) &= x_0, y(t_0) = y_0 \\ x(t_f) &= x_f, y(t_f) = y_f \\ \|y(t) - x(t)\|^2 &\geq d^2, \forall t \in [t_0, t_f] \\ \text{Minimiser } J &= \int_{t_0}^{t_f} \frac{u(t)^2 + v(t)^2}{2} dt \end{aligned}$$

Condition nécessaire de solution

Conformément à la condition nécessaire de solution rappelées en annexe A, nous pouvons, dans ce problème de commande optimale avec contrainte sur l'état, introduire la fonction $S(x, y) = \frac{d^2 - \|y - x\|^2}{2}$ que nous dériverons par rapport au temps jusqu'à ce qu'apparaisse explicitement la commande :

$$\dot{S}(x, y, t) = -(y - x)(v - u)$$

2.1. RÉOLUTION DE CONFLITS : PROBLÈME DE CONTRÔLE OPTIMAL 33

L'Hamiltonien augmenté d'ordre 1 du problème s'écrit alors :

$$H(x, y, u, v, t, p, q, l) = \lambda_0 \frac{u^2 + v^2}{2} + pu + qv - l(y - x).(v - u)$$

On supposera que l (notée μ dans l'annexe A) est telle que $l_3 = 0$ et que l_2 ne possède qu'un nombre fini de masses ponctuelles.

On peut alors écrire que la solution optimale du problème doit vérifier $H_u = 0$ et $H_v = 0$ d'une part et d'autre part $\dot{p} = -H_x$ et $\dot{q} = H_y$.

Si $\lambda_0 = 0$, alors, l'Hamiltonien devient :

$$H(x, y, u, v, t, p, q, l) = pu + qv - l(y - x).(v - u)$$

Si l'on pose $u(t) = -Cp(t)$ et $v(t) = -Cq(t)$ avec C constante fixée aussi grande qu'on veut, alors la minimisation de l'Hamiltonien ne permet pas de trouver une solution au problème, celui-ci n'ayant pas de borne inférieure. Or on suppose que le problème a une solution. Donc $\lambda_0 \neq 0$ et on prendra $\lambda_0 = 1$.

Les équations sont donc les suivantes :

$$\begin{aligned} u + p + l(y - x) &= 0 \\ v + q - l(y - x) &= 0 \\ \dot{p} &= -l(v - u) \\ \dot{q} &= l(v - u) \end{aligned}$$

Il faut ajouter les conditions aux limites en $t = t_0$ et $t = t_f$.

Invariants :

On remarque immédiatement que :

$$\begin{aligned} \dot{p} + \dot{q} &= 0 \\ u + v + p + q &= 0 \end{aligned}$$

On en déduit que $p + q = Cte$ et $u + v = Cte$, ceci pour tout $t \in [t_0, t_f]$.

Posons $u + v = 2K$. On a alors :

$$\int_{t_0}^{t_f} 2K dt = x(t_f) + y(t_f) - x(t_0) - y(t_0) = 2K(t_f - t_0) = 2KT$$

On a donc :

$$\forall t \in [t_0, t_f], \frac{x(t) + y(t)}{2} = (t - t_0)K + \frac{x_0 + y_0}{2} = tK + Q$$

Autrement dit, la trajectoire du barycentre des deux avions affectés des coefficients $\frac{1}{2}$ et $\frac{1}{2}$ est complètement décrite.

Le deuxième invariant du problème peut s'écrire dans le référentiel lié au barycentre cité ci-dessus. On observe en effet que :

$$[(q - p) \wedge (y - x)]' = (\dot{q} - \dot{p}) \wedge (y - x) + (q - p) \wedge (v - u) = 0$$

d'où puisque

$$(v - u) \wedge (y - x) + (q - p) \wedge (y - x) = 0$$

$$(v - u) \wedge (y - x) = Cte$$

Ce résultat peut s'écrire dans le repère barycentrique des deux avions :

$$\forall t \in [t_0, t_f], u_g(t) \wedge x_g(t) + v_g(t) \wedge y_g(t) = Cte$$

Remarque : si l'on considère au lieu de 2 avions, n avions, on montre alors que l'isobarycentre de ces n avions a une trajectoire rectiligne, se déplace à vitesse constante et que dans le référentiel barycentrique :

$$\forall t \in [t_0, t_f], \sum_{i=1}^n u_{ig}(t) \wedge x_{ig}(t) = Cte$$

Cas sans conflit

Lorsque la contrainte d'état n'est pas saturée, les équations se simplifient et deviennent :

$$\begin{aligned} p &= Cte \\ q &= Cte \\ u &= -p = Cte \\ v &= -q = Cte \end{aligned}$$

En conséquence, à l'optimum, tant que les avions ne sont pas à la distance d l'un de l'autre, leurs trajectoires sont rectilignes et leurs vitesses constantes.

Les différentes phases des trajectoires

Le cas où la contrainte d'état n'est jamais saturée nous intéresse peu, la solution est alors triviale, chaque avion ayant une vitesse constante définie à partir de ses conditions aux limites t_0 et t_f . On supposera donc que la contrainte est saturée en un point de $[t_0, t_f]$ au moins.

Proposition : Si la contrainte est saturée en t_1 et en t_2 , alors elle est saturée sur tout l'intervalle $[t_1, t_2]$.

2.1. RÉOLUTION DE CONFLITS : PROBLÈME DE CONTRÔLE OPTIMAL 35

Démonstration : supposons qu'il existe $t_3 \in]t_1, t_2[$ tel que la contrainte ne soit pas saturée en t_3 . Posons :

$$\begin{aligned} T_i &= \{t \in [t_1, t_3] \mid \|y(t) - x(t)\|^2 = d^2\} \\ T_s &= \{t \in [t_3, t_2] \mid \|y(t) - x(t)\|^2 = d^2\} \end{aligned}$$

T_i et T_s sont fermés car l'application $t \rightarrow \|x(t) - y(t)\|^2 - d^2$ est continue et bornée. T_i et T_s sont donc compacts dans \mathfrak{R} . Il existe donc $t_i \in [t_1, t_3]$ et $t_s \in [t_3, t_2]$ tels que :

$$\begin{aligned} t_i &= \sup(T_i) \\ t_s &= \inf(T_s) \end{aligned}$$

On a $t_i < t_3 < t_s$, car $t_3 \notin T_i$ et $t_3 \notin T_s$. La contrainte est saturée en t_i et t_s . Par contre, la contrainte n'est pas saturée sur $]t_i, t_s[$: les deux avions ont donc des trajectoires rectilignes à vitesse constante sur cet intervalle de temps. \dot{x} et \dot{y} sont donc constants sur $]t_i, t_s[$. On en déduit que $\dot{x} - \dot{y}$ est constant sur $]t_i, t_s[$ et donc $x(t) - y(t)$ est de la forme $tA + B$ pour $t \in]t_i, t_s[$. On peut alors écrire :

$$\|y(t) - x(t)\|^2 - d^2 = A^2 t^2 + 2A.B t + B^2 - d^2$$

Supposons que $A \neq 0$. On a alors un polynôme du second degré qui admet pour racines t_i et t_s . Ce polynôme est positif à l'extérieur des racines (car $A^2 > 0$) et négatif entre les racines. En particulier, $\|y(t_3) - x(t_3)\|^2 - d^2 < 0$ d'où la contradiction.

Donc $A = 0$ et $\|y(t) - x(t)\|^2 - d^2 = B^2 - d^2 = Cte$. Puisque $\|x(t) - y(t)\|^2$ est continue sur $[t_i, t_s]$, elle est donc constante sur tout l'intervalle $[t_i, t_s]$, et donc $\|x(t_3) - y(t_3)\|^2 = \|x(t_i) - y(t_i)\|^2 = d^2$, d'où la contradiction.

On en conclut que les avions, supposés séparés à t_0 et t_f , parcourent trois phases à l'optimum : de t_0 à t_1 et de t_2 à t_f , les avions ont des vecteurs vitesse constants. Entre t_1 et t_2 , la contrainte est saturée.

Trajectoire optimale et valeur minimale en fonction de la géométrie du problème

Supposons connus, (ou fixés) t_1, t_2 et

$$\begin{aligned} x(t_1) &= x_1 & x(t_2) &= x_2 \\ y(t_1) &= y_1 & y(t_2) &= y_2 \end{aligned}$$

La trajectoire du point milieu de x et y est connue (rectiligne uniforme). Lorsque la contrainte est saturée, les avions sont distants de $\frac{d}{2}$ de ce point. Leurs trajectoires

peuvent être décrites en définissant un angle $\theta(t)$. A $t \in [t_1, t_2]$, on pose :

$$\begin{aligned}x_1(t) &= K_1(t - t_0) + Q_1 - \frac{1}{2}d \cos \theta(t) \\x_2(t) &= K_2(t - t_0) + Q_2 - \frac{1}{2}d \sin \theta(t) \\y_1(t) &= K_1(t - t_0) + Q_1 + \frac{1}{2}d \cos \theta(t) \\y_2(t) &= K_2(t - t_0) + Q_2 + \frac{1}{2}d \sin \theta(t)\end{aligned}$$

On a :

$$\begin{aligned}\dot{x}_1(t) &= K_1 + \frac{1}{2}d \dot{\theta} \sin \theta(t) \\ \dot{x}_2(t) &= K_2 - \frac{1}{2}d \dot{\theta} \cos \theta(t) \\ \dot{y}_1(t) &= K_1 - \frac{1}{2}d \dot{\theta} \sin \theta(t) \\ \dot{y}_2(t) &= K_2 + \frac{1}{2}d \dot{\theta} \cos \theta(t)\end{aligned}$$

Posons :

$$\dot{\theta} = \rho$$

On en déduit :

$$J_{[t_1, t_2]} = \int_{t_1}^{t_2} (K_1^2 + K_2^2 + \frac{1}{4}d^2 \rho^2) dt$$

On a alors un problème de commande optimale entre t_1 et t_2 et l'Hamiltonien s'écrit :

$$H = K_1^2 + K_2^2 + \frac{1}{4}d^2 \rho^2 + \lambda \rho$$

On en déduit :

$$\begin{aligned}\frac{1}{2}d^2 \rho + \lambda &= 0 \\ \dot{\lambda} &= 0\end{aligned}$$

et donc :

$$\begin{aligned}\rho &= \frac{\theta_2 - \theta_1}{t_2 - t_1} \\ \theta(t) &= \theta_1 + \rho(t - t_1)\end{aligned}$$

avec :

$$\begin{aligned}\theta(t_1) &= \theta_1 \\ \theta(t_2) &= \theta_2\end{aligned}$$

2.1. RÉOLUTION DE CONFLITS : PROBLÈME DE CONTRÔLE OPTIMAL 37

Finalement, à l'optimum :

$$J_{[t_1, t_2]} = (t_2 - t_1)(K_1^2 + K_2^2) + \frac{d^2(\theta_2 - \theta_1)^2}{4(t_2 - t_1)}$$

Puisque :

$$x_{11} = K_1(t_1 - t_0) + Q_1 - \frac{1}{2}d \cos \theta_1$$

$$x_{12} = K_2(t_1 - t_0) + Q_2 - \frac{1}{2}d \sin \theta_1$$

$$y_{11} = K_1(t_1 - t_0) + Q_1 + \frac{1}{2}d \cos \theta_1$$

$$y_{12} = K_2(t_1 - t_0) + Q_2 + \frac{1}{2}d \sin \theta_1$$

pour $t \in [t_0, t_1]$:

$$u_{11} = K_1 + \frac{y_{01} - x_{01} - d \cos \theta_1}{2(t_1 - t_0)}$$

$$u_{12} = K_2 + \frac{y_{02} - x_{02} - d \sin \theta_1}{2(t_1 - t_0)}$$

$$v_{11} = K_1 - \frac{y_{01} - x_{01} - d \cos \theta_1}{2(t_1 - t_0)}$$

$$v_{12} = K_2 - \frac{y_{02} - x_{02} - d \sin \theta_1}{2(t_1 - t_0)}$$

d'où :

$$J_{[t_0, t_1]} = (t_1 - t_0)(K_1^2 + K_2^2) + \frac{(y_{01} - x_{01} - d \cos \theta_1)^2 + (y_{02} - x_{02} - d \sin \theta_1)^2}{4(t_1 - t_0)}$$

de même :

$$J_{[t_2, t_f]} = (t_f - t_2)(K_1^2 + K_2^2) + \frac{(y_{f1} - x_{f1} - d \cos \theta_2)^2 + (y_{f2} - x_{f2} - d \sin \theta_2)^2}{4(t_f - t_2)}$$

On peut donc exprimer J :

$$\begin{aligned} J(t_1, t_2, \theta_1, \theta_2) &= T(K_1^2 + K_2^2) \\ &+ \frac{(y_{01} - x_{01} - d \cos \theta_1)^2 + (y_{02} - x_{02} - d \sin \theta_1)^2}{4(t_1 - t_0)} \\ &+ \frac{(y_{f1} - x_{f1} - d \cos \theta_2)^2 + (y_{f2} - x_{f2} - d \sin \theta_2)^2}{4(t_f - t_2)} \\ &+ \frac{d^2(\theta_2 - \theta_1)^2}{4(t_2 - t_1)} \end{aligned}$$

Il s'agit maintenant de rechercher les extrema de J et de vérifier ensuite que l'on a bien un minimum. Les dérivations partielles de J par rapport à $\theta_1, \theta_2, t_1, t_2$ donnent les 4 équations suivantes :

$$\begin{aligned}\frac{(y_{02} - x_{02}) \cos \theta_1 - (y_{01} - x_{01}) \sin \theta_1}{t_1 - t_0} &= \frac{-d(\theta_2 - \theta_1)}{t_2 - t_1} \\ \frac{(y_{f2} - x_{f2}) \cos \theta_2 - (y_{f1} - x_{f1}) \sin \theta_2}{t_f - t_2} &= \frac{d(\theta_2 - \theta_1)}{t_2 - t_1} \\ \frac{(y_{01} - x_{01} - d \cos \theta_1)^2 + (y_{02} - x_{02} - d \sin \theta_1)^2}{(t_1 - t_0)^2} &= \frac{d^2(\theta_2 - \theta_1)^2}{(t_2 - t_1)^2} \\ \frac{(y_{f1} - x_{f1} - d \cos \theta_2)^2 + (y_{f2} - x_{f2} - d \sin \theta_2)^2}{(t_f - t_2)^2} &= \frac{d^2(\theta_2 - \theta_1)^2}{(t_2 - t_1)^2}\end{aligned}$$

On en déduit :

$$\begin{aligned}(y_{01} - x_{01}) \cos \theta_1 + (y_{02} - x_{02}) \sin \theta_1 &= d \\ (y_{f1} - x_{f1}) \cos \theta_2 + (y_{f2} - x_{f2}) \sin \theta_2 &= d\end{aligned}$$

ou si $\vec{\theta}_1$ et $\vec{\theta}_2$ sont les vecteurs unités formant un angle θ_1 et θ_2 avec le premier vecteur de la base canonique $(\vec{i}, \vec{j}, \vec{k})$:

$$(\vec{y}_0 - \vec{x}_0) \cdot \vec{\theta}_1 = d \quad (2.1)$$

$$(\vec{y}_f - \vec{x}_f) \cdot \vec{\theta}_2 = d \quad (2.2)$$

On a donc deux valeurs possibles de θ_1 et deux valeurs possibles de θ_2 . Posons :

$$a_1 = [(\vec{y}_0 - \vec{x}_0) \wedge \vec{\theta}_1] \cdot \vec{k} \quad (2.3)$$

$$a_2 = [(\vec{y}_f - \vec{x}_f) \wedge \vec{\theta}_2] \cdot \vec{k} \quad (2.4)$$

On a :

$$a_1^2 = \|y_0 - x_0\|^2 - d^2$$

$$a_2^2 = \|y_f - x_f\|^2 - d^2$$

Si $a_1 = 0$ ou $a_2 = 0$, alors on retrouve que $t_1 = t_0$ ou $t_f = t_2$. Si $\theta_2 = \theta_1$, alors on retrouve $t_2 = t_1$. Sinon :

$$\frac{-a_2}{t_f - t_2} = \frac{d(\theta_2 - \theta_1)}{t_2 - t_1} = \frac{+a_1}{t_1 - t_0} = \frac{d(\theta_2 - \theta_1) + (a_1 - a_2)}{T}$$

De la dernière équation, on déduit que a_1 et a_2 sont de signes opposés et donc, d'après 2.1, 2.2, 2.3 et 2.4, il n'existe que deux couples de valeurs (θ_1, θ_2) . Pour chaque couple, on définit t_1 et t_2 :

$$t_1 = t_0 + \frac{a_1 T}{d(\theta_2 - \theta_1) + (a_1 - a_2)}$$

$$t_2 = t_f + \frac{a_2 T}{d(\theta_2 - \theta_1) + (a_1 - a_2)}$$

2.1. RÉOLUTION DE CONFLITS : PROBLÈME DE CONTRÔLE OPTIMAL 39

Deux solutions du problème sans contrainte sont caractérisées.

Remarque : la différence $\theta_2 - \theta_1$ n'est définie qu'à $2k\pi$ près. Compte tenu de la forme de J , on minimisera J pour $\theta_2 - \theta_1$ dans $[-\pi, \pi]$.

On a finalement :

$$J = T\|\vec{K}\|^2 + \frac{\|\vec{y}_0 - \vec{x}_0 - d\vec{\theta}_1\|^2}{4(t_1 - t_0)} + \frac{\|\vec{y}_f - \vec{x}_f - d\vec{\theta}_2\|^2}{4(t_f - t_2)} + \frac{d^2(\theta_2 - \theta_1)^2}{4(t_2 - t_1)} \quad (2.5)$$

$$= T\|\vec{K}\|^2 + \frac{[d(\theta_2 - \theta_1) + (a_1 - a_2)]^2}{4T} \quad (2.6)$$

Le minimum sans contrainte respecte la contrainte

Il suffit de vérifier qu'avec les deux couples $(\theta_1, \theta_2, t_1, t_2)$, les avions restent distants d'au moins d entre t_0 et t_1 , et entre t_2 et t_f . Cela revient à résoudre deux équations du second degré. Les équations 2.1 et 2.2 apportent la preuve souhaitée.

Pour obtenir le meilleur des deux minima possibles, il reste désormais à comparer les deux minima obtenus. Les deux solutions correspondent aux cas où l'avion x passe devant l'avion y et où l'avion y passe devant l'avion x .

Compte tenu de l'expression 2.6, il s'agit de comparer, dans les deux cas, les valeurs absolues de : $d(\theta_2 - \theta_1) + (a_2 - a_1)$. On sait déjà que :

$$a_1 - a_2 = \pm(\sqrt{\|y_0 - x_0\|^2 - d^2} + \sqrt{\|y_f - x_f\|^2 - d^2})$$

Dans la partie suivante, nous donnerons une interprétation géométrique aux valeurs de a_1, a_2, θ_1 et θ_2 qui permettent de déterminer graphiquement le meilleur minimum.

Remarque : discontinuité aux limites

On sait que la présence de contraintes sur l'état peut induire une discontinuité des fonctions p, q et l aux points limites de saturation de la contrainte, c'est à dire en t_1 et t_2 . Les conditions supplémentaires sur l'optimum exposées par Kreindler [E.K82], Bryson, Denham et Dreyfus [JWS63], nous permettent d'évaluer ces fonctions. On a $\forall t \in [t_0, t_1] \cup [t_2, t_f]$:

$$\begin{aligned} p(t) &= -u(t) \\ q(t) &= -v(t) \\ l(t) &= 0 \end{aligned}$$

Si l'on pose $e = \frac{(y-x)}{d}$, on a pour $t \in [t_1, t_2]$:

$$\begin{aligned} \dot{p} &= [(K + p) \cdot e] \dot{e} \\ p(t_2) &= -u(t_2) \\ \dot{q} &= [(K + q) \cdot e] \dot{e} \\ q(t_2) &= -v(t_2) \end{aligned}$$

Et $\forall t \in [t_1, t_2]$:

$$\begin{aligned} p(t) &= -u(t) + \frac{(t_2 - t)}{2} (y - x) \\ q(t) &= -v(t) + \frac{(t_2 - t)}{2} (x - y) \\ l(t) &= \frac{t_2 - t}{2} \end{aligned}$$

On observe donc bien la discontinuité de p , q , et l en t_1 , qui amorce le virage des avions à savoir :

$$\begin{aligned} p(t_1^+) - p(t_1^-) &= \frac{(t_2 - t_1)}{2} (x_1 - y_1) \\ q(t_1^+) - q(t_1^-) &= \frac{(t_2 - t_1)}{2} (y_1 - x_1) \\ l(t_1^+) - l(t_1^-) &= \frac{(t_2 - t_1)}{2} \end{aligned}$$

2.1.2 Analyse de la sensibilité aux paramètres géométriques

On a représenté figure (2.1) les vecteurs $\vec{\theta}_1$ et $\vec{\theta}_2$ tels que leurs produits scalaires avec $\vec{y}_0 - \vec{x}_0$ et $\vec{y}_f - \vec{x}_f$ valent d .

ϕ est l'angle orienté $\in [0, 2\pi[$ entre $\vec{y}_0 - \vec{x}_0$ et $\vec{y}_f - \vec{x}_f$. Les indices $+$ et $-$ correspondent aux solutions obtenues pour a_1 et a_2 positif et négatif respectivement.

Les quantités $d(\theta_{2-} - \theta_{1+}) + a_{1+} - a_{2-}$ et $d(\theta_{2+} - \theta_{1-}) + a_{1-} - a_{2+}$ que l'on veut comparer sont représentées par les longueurs des chemins AP_1P_2B et AQ_1Q_2B , où P_1P_2 et Q_1Q_2 sont des arcs de cercles de rayon d et de centre O .

On observe qu'il n'y a conflit que si $\theta_{2+} - \theta_{1-} > 0$ et $\theta_{2-} - \theta_{1+} < 0$.

On observe de plus l'égalité suivante :

$$(\theta_{2+} - \theta_{1-}) + (\theta_{2-} - \theta_{1+}) = 2(\phi - \pi)$$

et donc :

$$d(\theta_{2+} - \theta_{1-}) + a_{1-} - a_{2+} + d(\theta_{2-} - \theta_{1+}) + a_{1+} - a_{2-} = 2d(\phi - \pi)$$

Si $\phi > \pi$, alors le chemin le plus court pour aller de A à B passe par Q_1 et Q_2 et donc on retiendra le minimum faisant apparaître $d(\theta_{2+} - \theta_{1-}) + a_{1-} - a_{2+}$. y passe devant x .

Si $\phi < \pi$ alors le chemin le plus court pour aller de A à B passe par P_1 et P_2 et donc on retiendra le minimum faisant apparaître $d(\theta_{2-} - \theta_{1+}) + a_{1+} - a_{2-}$. x passe devant y .

Si $\phi = \pi$ alors les deux minima sont égaux.

Dans tous les cas, on a :

$$J = T\|K\|^2 + \frac{(d(\pi - |\phi - \pi|) + |a_1| + |a_2| - d \arctan \frac{|a_1|}{d} - d \arctan \frac{|a_2|}{d})^2}{4T}$$

2.1. *RÉSOLUTION DE CONFLITS : PROBLÈME DE CONTRÔLE OPTIMAL* 41

FIGURE 2.1 – Détermination du meilleur minimum en fonction des conditions initiales et finales.

et donc graphiquement :

$$J = T\|K\|^2 + \frac{(\min(AP_1P_2B, AQ_1Q_2B))^2}{4T}$$

Remarques

On observe que l'avion qui passe devant à l'optimum est celui qui serait passé avant l'autre au croisement des trajectoires non modifiées.

Si les trajectoires des avions n'avaient pas été modifiées alors :

$$J_0 = T\|K\|^2 + \frac{((y_f - x_f) - (y_0 - x_0))^2}{4T}$$

et donc graphiquement :

$$J_0 = T\|K\|^2 + \frac{AB^2}{4T}$$

On en déduit :

$$J - J_0 = \frac{(\min(AP_1P_2B, AQ_1Q_2B))^2 - AB^2}{4T}$$

On mesure donc graphiquement "l'allongement en énergie" causé par le conflit.

Conclusion

Nous avons donc déterminé analytiquement la solution optimale d'une résolution de conflit minimisant l'intégrale des vitesses au carré des deux avions.

Les avions, pour s'éviter, tournent autour de leur isobarycentre à vitesse constante, isobarycentre qui se déplace lui-même à vitesse constante.

La propriété remarquable que nous avons exploitée résulte de la mise en évidence de deux invariants. L'extension à n avions ($n > 2$) de ce problème fait toujours apparaître ces deux invariants :

- Le centre de gravité des n avions a un mouvement rectiligne et uniforme.
- Dans le référentiel barycentrique :

$$\forall t \in [t_0, t_f], \sum_{i=1}^n u_{ig}(t) \wedge x_{ig}(t) = Cte$$

L'utilisation de ces deux invariants pour résoudre le problème à n avions (pour $n > 2$) est nettement plus difficile.

Dans le cas de deux avions, il a été possible de donner une interprétation géométrique à la valeur du minimum de la fonction. Cette interprétation géométrique permet de déterminer immédiatement le meilleur minimum.

On remarquera dans notre exemple que l'optimum dépend de l'éloignement en temps des avions par rapport au point de croisement mais pas de la vitesse de chacun des avions.

Remarque : Conflit entre deux avions contraints en vitesse

Ce cas est beaucoup plus réaliste que le précédent. On minimisera cette fois l'allongement de la trajectoire. Aussi on prendra

$$J = (\vec{x}_f - \vec{x}(t_f))^2 + (\vec{y}_f - \vec{y}(t_f))^2$$

sous les contraintes de vitesse et de séparation suivantes :

$$\begin{aligned} \|\dot{\vec{x}}\| &= C_x \\ \|\dot{\vec{y}}\| &= C_y \\ \|\vec{x} - \vec{y}\| &\geq d \end{aligned}$$

La condition nécessaire de solution permet de décrire les différentes phases de trajectoires. Lorsque la contrainte de séparation n'est pas saturée, les trajectoires des avions sont rectilignes et uniformes, leurs caps sont modifiés uniquement lorsque la contrainte de séparation est saturée. Ce résultat reste valable pour le cas de n avions. La contrainte sur les vitesses ne permet pas d'aller plus loin dans la résolution analytique du conflit. De plus, lorsqu'un conflit à deux avions se présente, il n'est pas "rentable" en terme de coût pour le contrôleur aérien de dévier les deux avions. En conséquence, dans la partie suivante, la trajectoire d'un des deux avions sera privilégiée (elle sera maintenue rectiligne et uniforme).

2.2 Le cas de deux avions lorsque l'un est privilégié.

2.2.1 Structure de l'ensemble des trajectoires admissibles

On s'intéresse à un conflit entre deux avions notés x et y par abus d'écriture, la trajectoire de l'un n'étant pas déviée. Les deux avions sont supposés se déplacer dans le plan horizontal entre un instant t_0 et t_f . L'ensemble des trajectoires admissibles est l'ensemble des courbes f_x et f_y paramétrées par le temps, continues, rectilignes par morceaux, sans point double, d'extrémités fixées en t_0 et t_f et telles que :

$$\forall t \in [t_0, t_f], \quad \|f_x(t) - f_y(t)\| \geq d$$

Si l'un des avions a une trajectoire rectiligne uniforme, on associe un repère à cet avion, l'ensemble des trajectoires de l'autre avion est alors (voir figure 2.2) l'ensemble des courbes, reliant deux points fixes A_0 et A_f du plan, qui sont continues, rectilignes par morceaux, qui ne rencontrent pas le disque fermé centré en O et de rayon d .

On va montrer, en identifiant le plan des trajectoires au plan complexe, que cet ensemble n'est pas connexe.

Définissons pour cela les ensembles suivants :

- L : ensemble des fonctions continues de $[t_0, t_f]$ dans le plan complexe, rectilignes par morceaux.
- D : ensemble des fonctions de L valant A_0 en t_0 et A_f en t_f .
- E : ensemble des fonctions f de D telles que $\|f(t)\| \geq d$ pour tout $t \in [t_0, t_f]$.

E est donc l'ensemble des trajectoires admissibles. $(f_1, f_2) \in E^2$ sont dans la même composante connexe si et seulement si :

$$\exists \phi \in \mathcal{C}^0([0, 1], E) \mid \phi(0) = f_1, \phi(1) = f_2$$

où la distance sur E est :

$$d(f_1, f_2) = \sup_{t \in [t_0, t_f]} \|f_1[t] - f_2[t]\|$$

Nous introduisons une condition supplémentaire portant sur les fonctions de E : les courbes du plan complexe paramétrées par les fonctions de E (i.e. les trajectoires admissibles pour l'avion dévié dans le référentiel de l'avion non dévié) ne bouclent pas autour de l'origine. Cela revient à éliminer *a priori* la possibilité pour un avion d'atteindre une certaine position par rapport au deuxième, puis de tourner autour de lui avant de reprendre par rapport à lui la même position. En vue de l'application à des trajectoires réelles, on peut faire cette hypothèse sans perte significative de généralité. Nous supposons de plus que la norme de la vitesse des deux avions est bornée par un réel V_{max} . Ainsi la norme de la vitesse relative de l'avion dévié dans le référentiel lié à l'avion non dévié est bornée

FIGURE 2.2 – Application du théorème des résidus de Cauchy

par $2V_{max}$. Les courbes paramétrées par les fonctions de E sont donc contenues dans un compact de C , que nous noterons K . On peut définir K de la façon suivante : $K = \{z \in C \mid d \geq |z| \geq (OA_0 + 2V_{max} * (t_f - t_0))\}$.

Théorème 2.1 – *L'ensemble E est composé de deux composantes connexes.*

Démonstration :

On définit une courbe joignant A_0 et A_f , de sorte qu'elle correspond au support d'une trajectoire d'un avion dévié passant derrière l'avion non dévié, mais orientée de A_f à A_0 . Cette courbe de référence sera notée T_r . A chaque trajectoire admissible pour l'avion dévié, i.e. à chaque fonction f de E , on associe la courbe orientée du plan complexe paramétrée par f , et notée $T(f)$. On définit $I : E \rightarrow R$ par

$$I(f) = \int_{T(f) \cup T_r} \frac{dz}{z}$$

On a vu que pour $f \in E$, la courbe du plan complexe paramétrée par f , $T(f)$, est contenue dans K , compact de C , dans lequel la fonction $z \rightarrow z^{-1}$ est uniformément continue. La fonction I est donc continue en f . Pour $f \in E$, et avec les hypothèses faites ci-dessus, le théorème des résidus de Cauchy assure que $I(f)$ ne peut prendre que les valeurs 0 ou 1. En effet, on peut choisir T_r de telle sorte que pour toutes les trajectoires de l'avion dévié considérées f , la courbe correspondante $T(f)$ ne coupe T_r qu'en A_0 et A_f , et reste à droite de $T(f)$ (voir figure 2.2). $T(f) \cup T_r$ est alors une courbe simple, et $I(f)$ vaut alors 1 si l'origine est à l'intérieur de cette courbe, et 0 sinon. On note $E_1 = \{f \in E \mid I(f) = 1\}$ et $E_0 = \{f \in E \mid I(f) = 0\}$ et on va montrer que E_1 et E_0 sont les deux composantes

connexes de E . Considérons deux trajectoires admissibles pour l'avion dévié, f_1 et f_2 , et les courbes correspondantes T_1 et T_2 .

1. f_1 et f_2 sont dans la même composante connexe de E . Dans ce cas, on a bien $I(f_1) = I(f_2)$: en effet I est continue en f donc $I(f_1)$ et $I(f_2)$ sont dans la même composante connexe de l'image de E par I .
2. f_1 et f_2 sont dans deux composantes connexes différentes de E : on veut montrer que dans ce cas on a bien $I(f_1) \neq I(f_2)$. Supposons que l'on ait $I(f_1) = I(f_2)$. On a alors :

$$I(f_1) - I(f_2) = 0$$

c'est à dire :

$$\int_{T(f_1) \cup -T(f_2)} \frac{dz}{z} = 0$$

où $-T(f_2)$ est la courbe orientée de même support que $T(f_2)$, mais orientée en sens inverse. D'après la définition de E , les courbes $T(f)$, avec $f \in E$, sont constituées d'un nombre fini de segments de droites. Donc $T(f_1)$ et $T(f_2)$ ne se coupent qu'en un nombre fini de points. On suppose d'abord que $T(f_1) \cup -T(f_2)$ forme une boucle simple. Dans ce cas, le théorème des résidus assure que le pôle de $z \rightarrow z^{-1}$ n'est pas à l'intérieur de $T(f_1) \cup -T(f_2)$, et donc d'après la forme de K , que l'intérieur de $T(f_1) \cup -T(f_2)$ est simplement connexe, ce qui entraîne que f_1 et f_2 sont dans la même composante connexe de E , ce qui achève la démonstration.

Si $T(f_1) \cup -T(f_2)$ ne forme pas une boucle simple, elle forme une suite finie de boucles simples. On peut alors appliquer la même démonstration à chaque boucle.

Extension à n avions

Dans le cas où l'on s'intéresse à un conflit comprenant n avions, on se retrouve face à $\frac{n(n-1)}{2}$ conflits potentiels. L'ensemble des solutions admissibles contient donc $2^{\frac{n(n-1)}{2}}$ composantes connexes. Dans la pratique, compte tenu des performances des avions, toutes les composantes connexes n'ont pas besoin d'être explorées. Néanmoins, la présence théorique d'autant d'ensembles disjoints et l'impossibilité de connaître *a priori* lequel contient la solution optimale rend le problème fortement combinatoire.

Extension dans le plan vertical

L'étude de la structure de l'ensemble des solutions admissibles ne reste pas valable lorsqu'on étend le problème à la dimension verticale. Néanmoins, dans

la pratique, on préfère donner des déviations dans le plan horizontal aux avions stables en croisière, les déviations verticales étant plutôt réservées aux avions en montée ou en descente. En conséquence, lorsqu'un problème à n avions comporte k avions stables en croisière, l'ensemble des solutions admissibles conserve au minimum $2^{\frac{k(k-1)}{2}}$ composantes connexes. Le problème restera donc généralement très combinatoire.

2.2.2 Trajectoires optimales de l'avion dévié

On peut, lorsque l'on privilégie un des deux avions en lui assurant une trajectoire rectiligne uniforme jusqu'à son objectif, s'intéresser à l'allongement de la deuxième trajectoire engendré par le conflit. Afin de simplifier le calcul, nous supposons dans un premier temps que les deux avions volent à la même vitesse $v = 1$. Les deux avions parcourent deux diamètres d'un cercle de rayon 1 entre t_0 et t_f . Compte tenu des normalisations, d exprime donc le rapport de la norme de séparation sur la distance de début de résolution au croisement des trajectoires. Le point de conflit des deux avions est donc le centre du cercle. On se placera dans le repère de l'avion qui n'est pas dévié et on mesurera le retard x de l'avion dévié à son point d'arrivée prévu. Etudions le cas simple où les deux avions sont sur deux axes perpendiculaires, nous ferons ensuite varier l'angle d'incidence des trajectoires.

Dans le repère de l'avion non dévié, on peut (voir figure 2.3) représenter l'origine et la destination de l'avion dévié, ainsi que la zone interdite pour l'avion dévié, à savoir le cercle de centre l'origine et de rayon d .

On sait déjà qu'à l'optimum, l'avion dévié rejoint ce cercle en ligne droite en un point repéré par l'angle θ_1 , y reste jusqu'à θ_2 et ensuite rejoint sa destination en ligne droite.

Il est facile d'observer que θ_2 est tel que la trajectoire en sortie est tangente au cercle d'évitement. Plaçons nous pour cela dans le repère absolu (voir figure 2.4). Supposons que l'avion dévié rejoigne le cercle d'évitement en θ_1 donné (dans le repère absolu, ce cercle se déplace). Nous avons démontré précédemment que l'avion reste sur le cercle d'évitement jusqu'en θ_2 puis le quitte pour rejoindre en ligne droite son objectif. Notons $\theta_{2-}(\theta_1)$ et $\theta_{2+}(\theta_1)$ les points tels que la trajectoire en sortie est tangente au cercle d'évitement. θ_{2-} et θ_{2+} dépendent uniquement du choix de θ_1 . On observe alors que pour rejoindre θ_2 , l'avion dévié venant de θ_1 doit passer soit par θ_{2-} soit par θ_{2+} . L'inégalité triangulaire prouve alors qu'il minimise son temps pour rejoindre l'objectif si $\theta_2 = \theta_{2-}$ ou $\theta_2 = \theta_{2+}$ selon qu'il est précédemment passé par θ_{2-} ou θ_{2+} . Intuitivement, cela revient à rejoindre "au plus court" l'objectif, une fois l'obstacle contourné. La trajectoire de l'avion dévié est donc régulière en sortie.

La régularité de la trajectoire en entrée est moins facile à montrer (elle n'est pas observable graphiquement). Aussi, nous ne fixerons pas la valeur de θ_1 *a priori*. θ_2 peut prendre deux valeurs suivant que l'avion dévié passe devant ou derrière l'avion non dévié. Nous noterons θ_{2+} et θ_{2-} respectivement ces valeurs.

FIGURE 2.3 – Trajectoires de l'avion dévié dans le repère lié à l'avion non dévié.

FIGURE 2.4 – Trajectoires de l'avion dévié dans le repère absolu.

On peut calculer pour deux points ϕ_1 et ϕ_2 du cercle le temps de parcours de l'avion dévié sur le cercle de $\theta = \phi_1$ à $\theta = \phi_2$. Pour cela, on intègre, sur la portion de cercle comprise entre ϕ_1 et ϕ_2 , l'inverse de la vitesse $v(\theta)$ de l'avion dévié dans le repère lié à l'avion non dévié.

$$Tps(\phi_1, \phi_2) = \int_{\phi_1}^{\phi_2} \frac{d\theta}{v(\theta)}$$

avec

$$v(\theta) = 2 \sin(\theta)$$

Si l'on pose :

$$T(\phi) = \frac{d}{4} \log \frac{1 - \cos \phi}{1 + \cos \phi} = \frac{d}{2} \log \left(\tan\left(\frac{\phi}{2}\right) \right)$$

On obtient finalement que :

$$Tps(\phi_1, \phi_2) = T(\phi_2) - T(\phi_1)$$

Le temps pour aller de θ_1 à θ_2 s'écrit donc $T(\theta_2) - T(\theta_1)$. On peut écrire que l'avion dévié rejoint son objectif en un temps valant $2 + x$ (voir figure 2.3). Notons $g_1(\theta_1)$ le temps mis par l'avion dévié pour rejoindre le cercle au point θ_1 , depuis son origine à $t = t_0$ et $g_2(\theta_2)$ le temps qu'il met pour quitter le cercle en θ_2 et rejoindre sa destination finale.

Soient $f_1(\theta_1) = g_1(\theta_1) - T(\theta_1)$ et $f_2(\theta_2) = g_2(\theta_2) + T(\theta_2)$.
Alors :

$$\begin{aligned} 2 + x &= g_1(\theta_1) + T(\theta_2) - T(\theta_1) + g_2(\theta_2) \\ &= f_1(\theta_1) + f_2(\theta_2) \end{aligned}$$

d'où l'on tire que

$$f_1(\theta_1) = F(x) = 2 + x - T(\theta_2) - g_2(\theta_2) \quad (2.7)$$

dans les deux cas suivants :

- $\theta = \theta_{2+}$:

$$\begin{aligned} T(\theta_{2+}) &= \frac{d}{2} \log \frac{\sqrt{2 + 2x + x^2 - d^2} - 1}{1 + x - d} \\ g_2(\theta_{2+}) &= \frac{2 + 2x + x^2}{2 \left(1 + x + \frac{d}{\sqrt{2 + 2x + x^2 - d^2}} \right)} \end{aligned}$$

- $\theta = \theta_{2-}$

$$\begin{aligned} T(\theta_{2-}) &= \frac{d}{2} \log \frac{\sqrt{2 + 2x + x^2 - d^2} + 1}{1 + x - d} \\ g_2(\theta_{2-}) &= \frac{2 + 2x + x^2}{2 \left(1 + x - \frac{d}{\sqrt{2 + 2x + x^2 - d^2}} \right)} \end{aligned}$$

Dans les formules précédentes, on calcule $\tan(\frac{\theta}{2})$ à partir des équations suivantes qui découlent de la figure 2.3 :

$$\begin{aligned} \cos(\theta_{1-} + \frac{\pi}{4}) &= \frac{d}{\sqrt{2}} \\ \sin(\theta_{1-} + \frac{\pi}{4}) &= \frac{-\sqrt{2 - d^2}}{\sqrt{2}} \\ \cos(\theta_{1+} + \frac{\pi}{4}) &= \frac{d}{\sqrt{2}} \\ \sin(\theta_{1+} + \frac{\pi}{4}) &= \frac{\sqrt{2 - d^2}}{\sqrt{2}} \\ \cos(\theta_{2-} - \rho) &= \frac{d}{\sqrt{1 + (1 + x)^2}} \\ \sin(\theta_{2-} - \rho) &= \frac{\sqrt{1 + (1 + x)^2 - d^2}}{\sqrt{1 + (1 + x)^2}} \\ \cos(\theta_{2+} - \rho) &= \frac{d}{\sqrt{1 + (1 + x)^2}} \end{aligned}$$

$$\begin{aligned}\sin(\theta_{2+} - \rho) &= \frac{-\sqrt{2-d^2}}{\sqrt{1+(1+x)^2}} \\ \cos(\rho) &= \frac{-(1+x)}{\sqrt{1+(1+x)^2}} \\ \sin(\rho) &= \frac{1}{\sqrt{1+(1+x)^2}}\end{aligned}$$

Or, on a également :

$$\begin{aligned}f_1(\theta_1) &= g_1(\theta_1) - T(\theta_1) \\ &= \frac{2+d^2-2d(\cos(\theta_1)-\sin(\theta_1))}{2(1-d\cos(\theta_1))} - \frac{d}{4} \log \frac{1-\cos\theta_1}{1+\cos\theta_1}\end{aligned}$$

En étudiant les variations de $F(x)$ dans les deux cas (θ_{2+} et θ_{2-}), on montre que F est une fonction croissante de x (voir figure 2.5), donc l'allongement x est minimal lorsque $F(x) = f_1(\theta_1)$ est minimal. En étudiant les variations de f_1 , on montre que f_1 n'est pas définie pour $\theta_1 = 0$ et décroît de part et d'autre de cette valeur (voir figure 2.5). On en déduit que le minimum de f_1 se situe aux bornes de l'intervalle autorisé pour θ_1 , à savoir en θ_{1+} et en θ_{1-} (voir figure 2.3). La trajectoire de l'avion dévié, entre son point de départ et le cercle d'évitement, est donc tangente à ce dernier à l'optimum. On a donc :

$$\begin{aligned}\tan \frac{\theta_{1+}}{2} &= \frac{-1 + \sqrt{2-d^2}}{1+d} \\ \tan \frac{\theta_{1-}}{2} &= \frac{-1 - \sqrt{2-d^2}}{1+d}\end{aligned}$$

On peut remarquer que le point $\theta_1 = 0$ ne permet pas à l'avion dévié de rejoindre son objectif.

On connaît donc maintenant les deux trajectoires minimales de notre problème d'optimisation sous contrainte de vitesse. Ces deux trajectoires correspondent aux deux minima locaux de notre fonction coût dans l'ensemble des fonctions admissibles.

En remplaçant dans l'expression 2.7, $f(\theta_{1+})$, $f(\theta_{1-})$, $T(\theta_{2+})$, $T(\theta_{2-})$, $g_2(\theta_{2+})$, $g_2(\theta_{2-})$, par leurs valeurs en fonction de x , on obtient les équations implicites donnant les valeurs des optima :

$$\begin{aligned}2+x_+ &= \frac{1}{1-\frac{d}{\sqrt{2-d^2}}} + \frac{2+2x_++x_+^2}{2\left(1+x_++\frac{d}{\sqrt{2+2x_++x_+^2-d^2}}\right)} + \frac{d}{2} \log\left(\frac{(1+d)\left(\sqrt{2+2x_++x_+^2-d^2}-1\right)}{(1+x_+-d)\left(\sqrt{2-d^2}-1\right)}\right) \\ 2+x_- &= \frac{1}{1+\frac{d}{\sqrt{2-d^2}}} + \frac{2+2x_-+x_-^2}{2\left(1+x_- - \frac{d}{\sqrt{2+2x_-+x_-^2-d^2}}\right)} + \frac{d}{2} \log\left(\frac{(1+d)\left(\sqrt{2+2x_-+x_-^2-d^2}+1\right)}{(1+x_--d)\left(\sqrt{2-d^2}+1\right)}\right)\end{aligned}$$

FIGURE 2.5 – $d = 0.1$ - A gauche : $F(x)$ pour les deux valeurs de θ_2 - A droite : $f_1(\theta_1)$ en fonction de θ_1

Ceci nous définit des fonctions implicites $x_+(d)$ et $x_-(d)$. Si l'on pose $d = \frac{1}{t}$, on peut représenter les pourcentages d'allongements de trajectoires en fonction de l'anticipation de la résolution du conflit (voir figure 2.6). Pour mesurer l'anticipation, l'unité de temps est alors le temps mis par un avion pour parcourir la valeur d'une norme de séparation. L'allongement en pourcentage représente l'allongement de la trajectoire rapporté au temps d'anticipation. On observe que les pourcentages d'allongements de trajectoires restent très faibles tant que t est supérieur à 2. Dans notre exemple, la trajectoire optimale est toujours celle qui passe derrière l'avion non dévié.

2.2.3 Influence de l'angle d'incidence

On peut assez facilement étendre l'étude précédente au cas de deux avions convergent non plus à angle droit mais sous un angle quelconque (voir figure 2.7). Toujours en ne bougeant qu'un seul avion, on peut déterminer les deux équations implicites qui nous donnent l'allongement de trajectoire du deuxième avion dans le cas où celui-ci passe devant (indice +) ou derrière (indice -). Avec les mêmes notations que précédemment, et en notant ϕ l'angle d'incidence des deux avions, $r(\phi) = 2 \sin(\frac{\phi}{2})$ la distance à $t = t_0$ entre les deux avions, et $\alpha(\phi) = \frac{\phi - \pi}{2}$ l'angle à $t = t_0$ de la trajectoire non déviée de l'avion dévié dans le repère de l'avion non dévié (voir figure 2.7), on obtient les équations suivantes :

1. On rappelle que les avions parcourent des diamètres d'un cercle de rayon 1

FIGURE 2.6 – Comparaison des allongements minimaux (en %) pour t supérieur à 1.5 (l'unité de temps est le temps mis par un avion pour parcourir une séparation standard). L'angle d'incidence des trajectoires est $\frac{\pi}{2}$.

FIGURE 2.7 – Représentation des trajectoires d'évitement optimales dans le repère lié à l'avion non dévié.

$$\begin{aligned}
\tan\left(\frac{\theta_{1-}}{2}\right) &= \frac{-\sqrt{-d^2 + r^2} + r \sin(\alpha)}{d + r \cos(\alpha)} \\
\tan\left(\frac{\theta_{1+}}{2}\right) &= \frac{\sqrt{-d^2 + r^2} + r \sin(\alpha)}{d + r \cos(\alpha)} \\
\tan\left(\frac{\theta_{2-}}{2}\right) &= \frac{-\sqrt{-d^2 + r^2 + x^2 + 2rx \cos(\alpha)} + r \sin(\alpha)}{-d + x + r \cos(\alpha)} \\
\tan\left(\frac{\theta_{2+}}{2}\right) &= \frac{\sqrt{-d^2 + r^2 + x^2 + 2rx \cos(\alpha)} + r \sin(\alpha)}{-d + x + r \cos(\alpha)} \\
x_{1+} = x_{1-} &= \sqrt{r^2 - d^2} \\
x_{2+} = x_{2-} &= \sqrt{r^2 - d^2 + x^2 + 2rx \cos(\alpha)} \\
v_{1-} &= \frac{2 \left(\sqrt{-d^2 + r^2} \cos(\alpha) - d \sin(\alpha) \right)}{r} \\
v_{1+} &= \frac{2 \left(\sqrt{-d^2 + r^2} \cos(\alpha) + d \sin(\alpha) \right)}{r} \\
v_{2-} &= \frac{2 \left((x + r \cos(\alpha)) \sqrt{-d^2 + r^2 + x^2 + 2rx \cos(\alpha)} + dr \sin(\alpha) \right)}{r^2 + x^2 + 2rx \cos(\alpha)} \\
v_{2+} &= \frac{2 \left((x + r \cos(\alpha)) \sqrt{-d^2 + r^2 + x^2 + 2rx \cos(\alpha)} - dr \sin(\alpha) \right)}{r^2 + x^2 + 2rx \cos(\alpha)}
\end{aligned}$$

Les deux équations implicites deviennent :

$$\begin{aligned}
2 + x_- &= \frac{x_{1-}}{v_{1-}} + \frac{x_{2-}}{v_{2-}} + \frac{d}{2} \log\left(\frac{\tan\left(\frac{\theta_{2-}}{2}\right)}{\tan\left(\frac{\theta_{1-}}{2}\right)}\right) \\
2 + x_+ &= \frac{x_{1+}}{v_{1+}} + \frac{x_{2+}}{v_{2+}} + \frac{d}{2} \log\left(\frac{\tan\left(\frac{\theta_{2+}}{2}\right)}{\tan\left(\frac{\theta_{1+}}{2}\right)}\right)
\end{aligned}$$

On trouvera en annexe C les courbes donnant les représentations graphiques des allongements de trajectoires en fonction des angles d'incidence et du temps d'anticipation.

Enfin, on peut représenter en fonction de l'angle d'incidence et du rapport des vitesses des deux avions, l'allongement minimal obtenu, pour une anticipation de deux fois la norme de séparation (voir figure 2.8) ou pour une anticipation de cinq fois la norme de séparation (voir figure 2.9). On observe alors les situations les plus pénalisantes dans le cas où les vitesses sont voisines et les angles d'incidence faibles, ce qui reste conforme aux résultats précédents ainsi qu'aux résultats décrits par AERA 3.

FIGURE 2.8 – Allongements minimaux en % en fonction de l'angle d'incidence (variant de 0 à π) et du rapport des vitesses (variant de 0.3 à 1) pour une anticipation valant deux fois la norme de séparation.

FIGURE 2.9 – Allongements minimaux en % en fonction de l'angle d'incidence (variant de 0 à π) et du rapport des vitesses (variant de 0.3 à 1) pour une anticipation valant cinq fois la norme de séparation.

2.3 Les limites du modèle pour $n > 2$ avions

Pour un conflit à n avions avec ($n > 2$), nous avons pu obtenir les résultats théoriques suivants. A l'optimum :

- Tant que l'avion n'est pas en conflit, sa trajectoire est rectiligne uniforme.
- Si les avions ne sont pas contraints en vitesse, les conditions nécessaires du problème de Contrôle Optimal permettent de dégager les deux invariants suivants :

1. Le centre de gravité des n avions a un mouvement rectiligne et uniforme.
2. Dans le référentiel barycentrique :

$$\forall t \in [t_0, t_f], \sum_{i=1}^n u_{ig}(t) \wedge x_{ig}(t) = Cte$$

Ces deux invariants sont difficilement exploitables pour $n > 2$. De plus, l'hypothèse faite sur les vitesses des avions n'est pas réaliste. Lorsque l'on contraint les avions en vitesse, ces deux invariants changent de forme et deviennent inexploitable même pour $n = 2$.

Pour $n = 2$, l'ensemble des solutions admissibles contient deux composantes connexes disjointes. Dans le cas où l'on s'intéresse à un conflit comprenant n avions, l'ensemble des solutions admissibles peut contenir $2^{\frac{n(n-1)}{2}}$ composantes connexes disjointes. Pour $n = 6$, cela représente 32768 composantes connexes potentielles. Pour $n = 7$ ce nombre est supérieur à 2 millions. Pour 10 avions, il est supérieur à 30 mille milliards. La présence théorique d'autant d'ensembles disjoints et l'impossibilité de connaître *a priori* lequel contient la solution optimale rend le problème fortement combinatoire.

Nombre de ces composantes connexes sont vides. D'autres peuvent être éliminées facilement *a priori*. Les applications numériques qui vont suivre montrent cependant qu'avec un faible nombre d'avions, il peut rester beaucoup de composantes connexes sans qu'il soit facile de déterminer laquelle contient l'optimum global.

angle en deg	allt en %
20	2.24
25	1.85
30	1.52
45	0.73
90	0.22
180	0.09

TABLE 2.1 – Allongements moyens en pourcentages en fonction de l’angle d’incidence.

2.4 Approche numérique

Dans un premier temps, le but de ce chapitre est de donner, grâce à un algorithme d’optimisation locale, les solutions optimales de certains conflits à deux avions. Dans un deuxième temps, l’extension à plusieurs avions est étudiée afin d’observer pratiquement les difficultés rencontrées lors de l’application de méthodes locales.

L’algorithme d’optimisation locale utilisé (LANCELOT) est décrit en annexe B. Pour pouvoir l’appliquer, les trajectoires d’avions ont été discrétisées en 20 points. Les contraintes de séparation des avions sont appliquées en chaque point de discrétisation. Dans les exemples présentés, les avions sont toujours contraints en vitesse. La norme de séparation est de $8 NM$ et les avions parcourent $120 NM$ à une vitesse de $400 kt$.

2.4.1 Conflit à deux avions

La figure 2.11 donne les trajectoires optimales des paires d’avions pour différents angles d’incidence. On observera que plus l’angle d’incidence est faible, plus l’allongement moyen des trajectoires est important (voir tableau 2.1). La figure 2.10 représente le résultat obtenu par LANCELOT pour un angle d’incidence de 15 degrés. On observe alors qu’après 1000 itérations, l’algorithme se trouve bloqué dans un minimum local, un avion se mettant en attente en “zigzagant” pour laisser passer l’autre (comme nous l’avons vu précédemment, ceci n’est pas optimal). Il semble que le pas de discrétisation soit alors trop important. Pour 20 degrés, 509 itérations suffisent, pour 25 degrés, 337 itérations sont nécessaires alors que pour un angle supérieur à 30 degrés, le nombre d’itérations nécessaires reste toujours inférieur à 200 (quelques minutes de calcul sur *HP720*).

2.4.2 Conflit à 5 avions

Les figures 2.12 et 2.13 donnent sept résolutions différentes d’un conflit à cinq avions. 7 points de départ différents ont été donnés à LANCELOT pour la recherche de la solution optimale, donnant 7 solutions localement optimales situées

FIGURE 2.10 – Angle : 15 deg, allt moy : 2.54%.

dans des composantes connexes différentes. Il existe probablement bien plus que sept composantes connexes disjointes pour ce problème, mais le but de ce paragraphe n'est pas de les chercher toutes. Pour la figure 2.14, le pas de discrétisation a été augmenté, car l'on observait le même phénomène que pour la figure 2.10, l'algorithme restant bloqué dans un minimum local en raison d'un pas de discrétisation trop fort : un des avions se met en attente en zigzagant sur chacun des pas de discrétisation.

On remarquera que les solutions, apparemment très différentes, génèrent des allongements moyens très voisins.

On notera que le temps de calcul nécessaire à la convergence de LANCELOT est très long : pour les exemples à 5 avions présentés ci-dessous, cela représente plusieurs heures (en moyenne 5) sur Sparc 2. Envisager une résolution en temps réel avec ce type d'algorithme ne paraît pas raisonnable.

On peut néanmoins réduire considérablement le temps de calcul si l'on accepte de moduler la vitesse des avions. La figure 2.15 donne des trajectoires localement optimales pour le même conflit à 5 avions mais en acceptant de réduire la vitesse des avions jusqu'à 10% de la vitesse initiale. L'algorithme converge alors beaucoup plus rapidement (une dizaine de minutes est nécessaire) et la solution obtenue en terme d'allongement n'est pas mauvaise.

L'observation des coefficients de Lagrange associés aux contraintes sur les vitesses montre dans notre exemple que lorsque la vitesse peut être réduite de 10%, 45% des contraintes sont saturées, alors que lorsque la vitesse ne peut être réduite, 100% des contraintes sont saturées. On observe également en fin de convergence que les coefficients de Lagrange des contraintes saturées, lorsque la vitesse peut être réduite de 10%, valent en moyenne le double des coefficients de Lagrange des contraintes saturées lorsque la vitesse ne peut pas être réduite.

On observe donc à travers cet exemple que la contrainte sur les vitesses est une contrainte forte.

2.4.3 Remarques

Les applications numériques présentées dans cette partie conduisent à plusieurs remarques :

- Les conflits à angle d'incidence faible sont plus coûteux et plus difficiles à résoudre que les conflits à angle d'incidence fort.
- La contrainte sur les vitesses est une contrainte forte.
- La complexité de l'ensemble des trajectoires admissibles ne permet pas d'obtenir par une méthode locale de type gradient la solution optimale du problème mais seulement des solutions localement optimales dépendant du point de départ utilisé pour l'algorithme. *A priori*, il n'existe pas de méthode connue pour détecter le point de départ qui permette à un algorithme de type gradient d'atteindre la solution optimale.
- Les conflits à résoudre peuvent avoir des solutions très proches en terme d'allongement de trajectoires et cependant très différentes. Il paraît donc tout à fait intéressant dans un but opérationnel de ne pas rechercher la solution optimale, mais un ensemble de solutions différentes d'allongements voisins de l'optimum. Il est donc nécessaire de s'intéresser à des méthodes d'optimisation globale permettant éventuellement de trouver plusieurs optima.

Conclusion

La modélisation par Contrôle Optimal que nous avons abordée dans cette première partie permet de dégager certaines propriétés des trajectoires optimales et de la structure de l'ensemble des solutions admissibles.

Nous retiendrons tout d'abord que les trajectoires non contraintes sont rectilignes et uniformes à l'optimum. Dans la pratique, la durée de saturation des contraintes, compte tenu de la vitesse élevée des avions et de la faible densité de trafic, est très faible comparée à la durée de vol sans saturation de contrainte (dans la mesure où les conflits ne sont pas à angle trop faible). En conséquence, la plupart des trajectoires optimales peuvent être approchées par des trajectoires rectilignes par morceaux (c'est l'objet du prochain chapitre). Dans le cas où les avions ne sont pas contraints en vitesse, on a montré que l'on peut déterminer géométriquement la solution optimale. Rien ne permet de penser que c'est encore le cas pour des avions contraints en vitesse.

Le second point important résulte de la structure de l'ensemble des solutions admissibles, celui-ci pouvant contenir pour n avions $2^{\frac{n(n-1)}{2}}$ composantes connexes disjointes. Le problème est de ce fait fortement combinatoire et nous avons pu observer numériquement qu'un algorithme d'optimisation locale tel que LANCELOT ne permet pas de trouver l'optimum global pour des valeurs de n faibles (par

FIGURE 2.11 – De haut en bas et de gauche à droite, angle : 20, 25, 30, 45, 90 et 180 degrés.

FIGURE 2.12 – De haut en bas et de gauche à droite, allongements moyens : 2.16, 2.21, 2.22, 2.20, 2.19 et 2.17 %.

FIGURE 2.13 – Allongement moyen : 2.22%.

FIGURE 2.14 – Discrétisation : à gauche 40 points, à droite 60 points.

FIGURE 2.15 – La vitesse peut être réduite jusqu'à 10%, allt moy : 2.35%.

exemple $n = 5$). Ce second point nous conduira dans le chapitre 4 à utiliser des algorithmes d'optimisation globale tels que les algorithmes génétiques.

Chapitre 3

Modélisations simplifiées : point tournant et “offset”

Introduction

Nous avons pu mettre en évidence, dans le chapitre précédent, la structure des trajectoires optimales pour des conflits à deux avions dans le plan horizontal. Dans la mesure où la durée de la saturation des contraintes est faible par rapport à la durée de vol sans saturation, on peut approcher les trajectoires des avions par des trajectoires rectilignes par morceaux. Ceci nous amène à introduire dans la première partie l’approximation “point tournant” et dans la deuxième partie la modélisation “offset” ou mise en route parallèle qui a l’avantage de linéariser dans une certaine mesure les contraintes. La partie trois compare les deux modélisations dans le cas général. Dans la partie quatre, nous nous intéressons au cas des avions en rattrapage pour lesquels la modélisation par offset est très avantageuse. La partie cinq s’intéresse aux possibilités de résolution en vitesse qui n’ont pas été abordées jusque-là. Dans la partie six, l’incertitude sur les vitesses et sur les taux de montée et de descente est modélisée de façon à pouvoir rendre réaliste et robuste les résolutions de conflits. La dernière partie est consacrée à l’évolution de la modélisation dans le cadre d’une optimisation en temps réel.

3.1 Approximation par point tournant

Les trajectoires optimales étudiées dans la section 2.2 étant formées de deux segments rectilignes en général relativement longs, et d’une portion courbe réduite, il apparaît naturel de vouloir approcher les trajectoires obtenues par des trajectoires composées de deux droites se coupant en un point tournant (figure 3.1). L’objet de ce paragraphe est d’évaluer l’allongement supplémentaire engendré par cette approximation. Il est intéressant de noter qu’on réduit alors l’ensemble de fonctions admissibles à un sous-ensemble de \mathcal{R}^2 , ou de \mathcal{R}^4 si l’on s’autorise à modifier la trajectoire de chaque avion.

FIGURE 3.1 – Modélisation point tournant.

3.1.1 Calcul préliminaire

Reprenons l'exemple traité section 2.2, et désignons dans le repère de l'avion non dévié par x_{1-} la distance de l'origine à θ_{1-} de l'avion dévié, par v_{1-} sa vitesse sur ce tronçon, et par l_{-} , la distance supplémentaire entre θ_{1-} et le point tournant (idem pour θ_{2-}).

$$\begin{aligned} \tan\left(\frac{\theta_{1-}}{2}\right) &= -\frac{1 + \sqrt{2 - d^2}}{1 + d} \\ \tan\left(\frac{\theta_{2-}}{2}\right) &= -\frac{1 + \sqrt{2 - d^2 + 2x_- + x_-^2}}{1 - d + x_-} \\ v_{1-} &= d + \sqrt{2 - d^2} \\ v_{2-} &= \frac{2 \left(-d + (1 + x_-) \sqrt{2 - d^2 + 2x_- + x_-^2} \right)}{2 + 2x_- + x_-^2} \\ x_{1-} &= \sqrt{2 - d^2} \\ x_{2-} &= \sqrt{2 - d^2 + 2x_- + x_-^2} \\ l_{-} &= \left\| \frac{d \left(\tan\left(\frac{\theta_{2-}}{2}\right) - \tan\left(\frac{\theta_{1-}}{2}\right) \right)}{1 + \tan\left(\frac{\theta_{2-}}{2}\right) \tan\left(\frac{\theta_{1-}}{2}\right)} \right\| \end{aligned}$$

On obtient alors l'équation implicite donnant l'allongement x_- :

$$2 + x_- = \frac{x_{1-} + l_-}{v_{1-}} + \frac{x_{2-} + l_-}{v_{2-}}$$

De même pour x_+ , on a :

$$\begin{aligned} \tan\left(\frac{\theta_{1+}}{2}\right) &= \frac{-1 + \sqrt{2 - d^2}}{1 + d} \\ \tan\left(\frac{\theta_{2+}}{2}\right) &= \frac{-1 + \sqrt{2 - d^2 + 2x_+ + x_+^2}}{1 - d + x_+} \\ v_{1+} &= -d + \sqrt{2 - d^2} \\ v_{2+} &= \frac{2 \left(d + (1 + x_+) \sqrt{2 - d^2 + 2x_+ + x_+^2} \right)}{2 + 2x_+ + x_+^2} \\ x_{1+} &= \sqrt{2 - d^2} \\ x_{2+} &= \sqrt{2 - d^2 + 2x_+ + x_+^2} \\ l_+ &= \left\| \frac{d \left(\tan\left(\frac{\theta_{2+}}{2}\right) - \tan\left(\frac{\theta_{1+}}{2}\right) \right)}{1 + \tan\left(\frac{\theta_{2+}}{2}\right) \tan\left(\frac{\theta_{1+}}{2}\right)} \right\| \end{aligned}$$

On obtient de même l'équation implicite donnant l'allongement x_+ :

$$2 + x_+ = \frac{x_{1+} + l_+}{v_{1+}} + \frac{x_{2+} + l_+}{v_{2+}}$$

3.1.2 Représentation graphique

On peut observer l'augmentation en pourcentage de l'allongement lorsque l'on remplace la trajectoire d'évitement réelle par un point tournant (voir figure 3.2). On observe que pour la solution optimale, l'augmentation d'allongement due à notre approximation reste inférieure à 0.25% tant que le temps d'anticipation t (t est le rapport entre la distance au point de conflit et la séparation standard d) est supérieur à 2 et que pour l'autre solution, cet écart ne dépasse pas 3%.

3.1.3 Influence de l'angle d'incidence

On peut, grâce aux calculs précédents représenter les surfaces définies par $x_-(d, \phi)$, $x_{t-}(d, \phi)$, $x_+(d, \phi)$, $x_{t+}(d, \phi)$ et $x_{t-}(d, \phi) - x_-(d, \phi)$, $x_{t+}(d, \phi) - x_+(d, \phi)$ (voir figures C.1, C.2 et C.3, annexe C).

On peut ainsi observer figure 3.3 les couples (t, Φ) pour lesquels l'allongement supplémentaire induit par l'approximation point tournant est inférieur à 5%.

On trouvera en annexe D diverses représentations de l'ensemble des points tournants admissibles dans \mathbb{R}^4 ainsi que les positions des optima. Ces représentations permettent d'observer dans la plus part des cas les deux composantes connexes

FIGURE 3.2 – Allongements en %, l'avion dévié passant derrière (à gauche) ou devant (à droite) l'avion non dévié, l'unité en abscisse est le temps mis par un avion pour parcourir une séparation standard d . L'angle d'incidence des trajectoires est de $\frac{\pi}{2}$.

FIGURE 3.3 – Augmentation de l'allongement $x_{t+} - x_+$ en fonction de t (variant de 0.5 à 5) et ϕ (variant de 0 à π).

évoquées dans la partie 2.2.1. On trouvera également dans cette annexe la représentation du coût de résolution de conflit en fonction de l'angle d'incidence et du temps d'anticipation de conflit, représentation conforme aux résultats décrits dans le projet AERA3 [Nie89b] lorsqu'un seul avion est dévié.

FIGURE 3.4 – Modélisation de l’offset.

3.2 Approximation par offset

3.2.1 Définition de l’offset

On appellera résolution par offset, la solution de notre programme de minimisation lorsque l’on met en parallèle l’un des deux avions impliqués dans le conflit. C’est la solution adoptée par Niedringhaus [NFC⁺83, Nie89b, Nie89a] pour le projet d’automatisation complète AERA 3 dans la fonction “Gentle-Strict”. Le principal intérêt de cette modélisation est que la fonction à minimiser devient linéaire, de même que les contraintes de séparation des avions une fois l’offset réalisé (à condition que l’angle de déviation soit fixé *a priori*). Nous verrons cependant qu’un certain nombre de contraintes non linéaires doivent être vérifiées *a posteriori* et que l’extension au cas de plus de deux avions de ce programme de minimisation n’est pas évidente.

3.2.2 Problème linéaire avec contraintes linéaires

Reprenons le cas de nos deux avions volant à vitesse constante, et supposons que leurs trajectoires se coupent à l’origine de notre repère (voir figure 3.4). On appellera ϕ l’angle d’incidence des deux avions, v_x et v_y , les vitesses respectives de l’avion x et de l’avion y , t_x et t_y , les temps respectifs au conflit des avions x et y à $t = 0$. On désignera par x la valeur absolue de l’offset pour l’avion x en terme de distance.

Supposons que les deux avions suivent leurs trajectoires non déviées. Pour vé-

rier la contrainte de séparation, on paramétrise les trajectoires dans le temps et on vérifie que la distance au carré entre les deux avions moins la séparation standard au carré d^2 reste positive. Il s'agit de vérifier que le discriminant d'un polynôme du second degré est négatif.

On obtient la condition nécessaire et suffisante suivante :

$$(t_x - t_y)^2 v_x^2 v_y^2 \sin(\phi)^2 \geq d^2 (v_x^2 + v_y^2 - 2 v_x v_y \cos(\phi))$$

On peut écrire cette condition sous la forme de deux équations linéaires :

$$(t_x - t_y) v_x v_y \sin(\phi) \geq \left(d \sqrt{v_x^2 + v_y^2 - 2 v_x v_y \cos(\phi)} \right)$$

ou bien

$$(t_y - t_x) v_x v_y \sin(\phi) \geq \left(d \sqrt{v_x^2 + v_y^2 - 2 v_x v_y \cos(\phi)} \right)$$

Supposons donc maintenant que l'avion x se mette en offset avec un angle β à une distance x de sa trajectoire et intéressons nous à la nouvelle condition de séparation à réaliser, une fois l'avion établi sur sa route parallèle. Le problème est en fait le même que le précédent si l'on modifie t_x et t_y de la façon suivante :

- Il faut ajouter à t_x le retard entraîné par la mise en offset, à savoir : $\frac{x \tan(\frac{\beta}{2})}{v_x}$.
- A cela s'ajoute ou se retranche, selon le sens de la déviation, le retard ou l'avance entraîné par le déplacement de la trajectoire de l'avion x , à savoir : $\frac{x \cot(\phi)}{v_x}$.
- Il faut ajouter ou retrancher à t_y , selon le sens de la déviation, le retard ou l'avance entraîné par le déplacement de la trajectoire de l'avion x , à savoir : $\frac{x}{\sin(\phi) v_y}$.

Les deux sens de déviations possibles, conjugués aux deux conditions citées précédemment nous conduisent à rechercher le minimum de la déviation de x sous la contrainte $x \geq 0$ dans les quatre cas suivant :

- déviation extérieure (sens de la figure 3.4) et l'avion x passant derrière :

$$\left(t_x + \frac{x \tan(\frac{\beta}{2})}{v_x} + \frac{x \cot(\phi)}{v_x} - t_y - \frac{x}{\sin(\phi) v_y} \right) v_x v_y \sin(\phi) \geq \left(d \sqrt{v_x^2 + v_y^2 - 2 v_x v_y \cos(\phi)} \right)$$

- déviation intérieure (sens inverse de la figure 3.4) et l'avion x passant derrière :

$$\left(t_x + \frac{x \tan(\frac{\beta}{2})}{v_x} - \frac{x \cot(\phi)}{v_x} - t_y + \frac{x}{\sin(\phi) v_y} \right) v_x v_y \sin(\phi) \geq \left(d \sqrt{v_x^2 + v_y^2 - 2 v_x v_y \cos(\phi)} \right)$$

- déviation extérieure (sens de la figure 3.4) et l'avion x passant devant :

$$\left(t_y + \frac{x}{\sin(\phi) v_y} - t_x - \frac{x \tan(\frac{\beta}{2})}{v_x} - \frac{x \cot(\phi)}{v_x} \right) v_x v_y \sin(\phi) \geq \left(d \sqrt{v_x^2 + v_y^2 - 2 v_x v_y \cos(\phi)} \right)$$

- déviation intérieure (sens inverse de la figure 3.4) et l'avion x passant devant :

$$\left(t_y - \frac{x}{\sin(\phi) v_y} - t_x - \frac{x \tan(\frac{\beta}{2})}{v_x} + \frac{x \cot(\phi)}{v_x} \right) v_x v_y \sin(\phi) \geq \left(d \sqrt{v_x^2 + v_y^2} - 2 v_x v_y \cos(\phi) \right)$$

Il suffit donc de retenir le plus petit des x obtenus (deux de ces contraintes sont incompatibles avec $x \geq 0$) pour avoir la déviation minimale. On observe évidemment que le coût de calcul est très faible car il suffit de résoudre une équation du type $ax = b$ quatre fois, et de choisir le plus petit des deux x positifs obtenus. On remarquera que l'offset obtenu ne dépend que du rapport $\frac{t_x}{t_y}$.

3.2.3 Généralisation

On peut bien évidemment autoriser l'avion y à réaliser un offset, et l'on minimise alors $x + y$, avec $x \geq 0$ et $y \geq 0$ sous huit contraintes linéaires en x et y différentes. Cela reste très simple et le minimum obtenu ne dévie qu'un seul avion (on minimise en effet une fonction linéaire avec des contraintes linéaires, la solution est donc sur une arête de l'ensemble des solutions admissibles).

Cette modélisation étendue au cas de n avions nécessite la résolution de $2^{\frac{n(n+1)}{2}}$ programmes linéaires comprenant chacun $\frac{n(n+1)}{2}$ contraintes linéaires. Pour un conflit à 5 avions, cela représente 32768 programmes d'optimisation à résoudre comprenant 15 contraintes linéaires chacun. Pour $n = 7$, plus de 2 millions de programmes linéaires doivent être résolus, comprenant 21 contraintes chacun. On se retrouve face à une très forte combinatoire. Il est intéressant de noter que l'on retrouve les composantes connexes observées dans le paragraphe 2.2.1. En effet, chaque programme linéaire détermine un ordre de passage entre une paire d'avions et résout le problème à l'intérieur d'une composante connexe de l'ensemble des solutions admissibles. Il faut néanmoins choisir le sens de la déviation pour chaque avion pour que le programme devienne linéaire ; c'est pourquoi la complexité passe de $2^{\frac{n(n-1)}{2}}$ à $2^{\frac{n(n-1)}{2}} \times 2^n = 2^{\frac{n(n+1)}{2}}$. F. Médioni [MDA94] s'est intéressé à combiner un algorithme génétique avec un programme d'optimisation linéaire pour résoudre le problème de résolution de conflits linéarisé. Les résultats pour des cas à cinq avions sont encourageants.

3.2.4 Contraintes supplémentaires non-linéaires

Nous nous sommes intéressés jusqu'à présent à satisfaire la contrainte de séparation lorsque l'avion est établi en offset. Durant la mise en offset, les deux avions doivent rester séparés.

Pour cela, nous allons définir une fonction qui détermine la séparation des avions lorsque ceux-ci parcourent à vitesse constante un segment de droite.

Supposons qu'à t_1 , l'avion x ait pour coordonnées x_{o1} et x_{o2} , et qu'à t_2 , ses coordonnées soient x_{d1} et x_{d2} . Prenons des notations similaires pour l'avion y , en

remplaçant x par y dans les notations. Enfin, on suppose que les avions x et y ne sont pas en évolution entre t_1 et t_2 .

Posons :

$$\begin{aligned}
 xy_{o1} &= y_{o1} - x_{o1} \\
 xy_{o2} &= y_{o2} - x_{o2} \\
 xy_{d1} &= y_{d1} - x_{d1} \\
 xy_{d2} &= y_{d2} - x_{d2} \\
 q_1 &= xy_{o1} - xy_{d1} \\
 q_2 &= xy_{o2} - xy_{d2} \\
 det &= q_1 \cdot xy_{d2} - q_2 \cdot xy_{d1}
 \end{aligned}$$

En paramétrant à nouveau les segments, on cherche le minimum de la distance au carré entre les deux avions, minimum qui peut se trouver à l'une des deux extrémités, ou au milieu (on vérifie alors qu'un certain discriminant est négatif). On montre alors que les avions restent séparés de d sur le segment si et seulement si :

- Si $q_1 \cdot xy_{d1} + q_2 \cdot xy_{d2} \geq 0$, alors $(xy_{d1})^2 + (xy_{d2})^2 \geq d^2$.
- Si $q_1 \cdot xy_{o1} + q_2 \cdot xy_{o2} \leq 0$, alors $(xy_{o1})^2 + (xy_{o2})^2 \geq d^2$.
- Dans les autres cas, $(det)^2 \geq d^2((q_1)^2 + (q_2)^2)$

Dans notre cas, nous poserons :

$$\begin{aligned}
 x_{o1} &= -v_x t_x \\
 x_{o2} &= 0 \\
 x_{d1} &= -v_x t_x + x \cot \beta \\
 x_{d2} &= \perp x \\
 y_{o1} &= -v_y t_y \cos \phi \\
 y_{o2} &= -v_y t_y \sin \phi \\
 y_{d1} &= -v_y \left(t_y - \frac{x}{v_x \sin \beta} \right) \cos \phi \\
 y_{d2} &= -v_y \left(t_y - \frac{x}{v_x \sin \beta} \right) \sin \phi
 \end{aligned}$$

Si la valeur de l'offset est indépendante du rapport $\frac{t_x}{t_y}$, les temps d'anticipation t_x et t_y doivent cependant permettre à l'avion dévié de se mettre en offset sans générer de conflit. Pour déterminer le moment de mise en offset le plus tardif, il faudra donc minimiser t_x (si l'avion x est dévié) sous la contrainte de séparation non-linéaire citée ci-dessus. De même, on devra s'assurer que le retour de l'avion dévié sur sa trajectoire s'effectue sans générer de conflit. On peut bien évidemment se libérer de ces contraintes en déviant les avions suffisamment tôt et en les ramenant suffisamment tard sur leur trajectoire, ou en imposant qu'à $t = 0$ et $t = t_f$ les avions soient loin les uns des autres. Mais dans la perspective d'une résolution automatique de conflits avec beaucoup d'avions, et des situations difficiles, ces hypothèses sont sans doute trop restrictives.

3.2.5 Résultats numériques

Nous avons représenté dans les cas suivants :

- l'avion dévié passant devant l'avion non dévié
- l'avion dévié passant derrière l'avion non dévié

avec $v_x = v_y = 1$, une norme de séparation $d = 1$, les pourcentages d'allongements des trajectoires en fonction du temps d'anticipation $tx = ty$ (l'unité de temps est toujours le temps mis par un avion pour parcourir une séparation standard) pour des angles d'incidences de trajectoires valant $\frac{\pi}{2}$ (figure 3.5), $\frac{7\pi}{8}$ (figure 3.6) et $\frac{\pi}{8}$ (figure 3.7), et des angles de mise en offset valant $\frac{\pi}{3}$, $\frac{\pi}{4}$, $\frac{\pi}{6}$, $\frac{\pi}{8}$. Pour une incidence de trajectoires valant $\frac{\pi}{8}$, l'avion dévié ne peut pas passer devant l'avion non dévié avec ce type d'offset.

Lorsque le temps d'anticipation est trop court, la courbe est ramenée à 0. Nous avons conservé sur les représentations graphiques les allongements optimaux obtenus dans le paragraphe précédent.

On remarquera que si la modélisation par offset est bonne lorsque l'angle d'incidence se rapproche de π , elle est nettement moins bonne lorsque l'angle d'incidence se rapproche de 0.

3.2.6 Contrainte sur l'angle de manœuvre de l'avion dévié

De même que l'on peut effectuer une mise en offset sous divers angles, on peut s'intéresser à limiter l'angle de manœuvre de l'avion dévié dans la modélisation point tournant. Notons β l'angle de déviation maximal que l'on autorise à l'avion dévié. Dans la modélisation point tournant, l'avion dévié l'est trois fois. Si l'on limite à β la première et la dernière déviation, la deuxième est nécessairement limitée à 2β . Cependant, afin de pouvoir comparer les modélisations point tournant et offset, nous ne limiterons que la première et la dernière déviation. En effet, dans la modélisation par offset, l'avion est dévié quatre fois, et le cumul des deux déviations intermédiaires vaut 2β . Il est donc toujours supérieur à la déviation intermédiaire de la modélisation point tournant lorsque la première et la dernière déviation sont inférieures à β .

Notons μ_- et μ_+ les bornes inférieures et supérieures de l'angle ρ d'incidence de la trajectoire de l'avion dévié dans le repère lié à l'avion non dévié (voir figure 3.8). μ_- , μ_+ , α et β sont liés par les équations :

$$\begin{aligned}\cos\mu_- &= \left| \sin \frac{\phi + \beta}{2} \right| \\ \cos\mu_+ &= \left| \sin \frac{\phi - \beta}{2} \right|\end{aligned}$$

De plus, reprenant les calculs du paragraphe précédent, on peut exprimer ρ en fonction de d , l'allongement x et l'angle d'incidence ϕ . Nous noterons ρ_{1-} l'angle de la trajectoire par derrière dans la première branche dans le repère lié à l'avion non dévié. L'indice 2 sera utilisé pour la deuxième branche, et l'indice + pour les

FIGURE 3.5 – Allongements optimaux en %, l'avion dévié passant derrière (à gauche) ou devant (à droite) l'avion non dévié pour $\phi = \frac{\pi}{2}$, en fonction du temps d'anticipation (l'unité de temps est le temps mis par les avions pour parcourir une séparation standard).

FIGURE 3.6 – Allongements optimaux en %, l'avion dévié passant derrière (à gauche) ou devant (à droite) l'avion non dévié pour $\phi = \frac{7\pi}{8}$, en fonction du temps d'anticipation (l'unité de temps est le temps mis par les avions pour parcourir une séparation standard).

FIGURE 3.7 – Allongements optimaux en %, l'avion dévié passant derrière (à gauche) ou devant (à droite) l'avion non dévié pour $\phi = \frac{\pi}{8}$, en fonction du temps d'anticipation (l'unité de temps est le temps mis par les avions pour parcourir une séparation standard).

FIGURE 3.8 – Contrainte sur l’angle de déviation, schéma explicatif.

trajectoires par devant.

$$\begin{aligned}\rho_{1-} &= \frac{-\theta_{1-} - \pi}{2} \\ \rho_{1+} &= \frac{-\theta_{1+} + \pi}{2} \\ \rho_{2-} &= \frac{-\theta_{1-} - \pi}{2} \\ \rho_{2+} &= \frac{-\theta_{2+} + \pi}{2}\end{aligned}$$

Il reste à vérifier que ρ est compris entre μ_- et μ_+ pour chaque cas pour être assuré que l’angle de déviation de la trajectoire ne dépasse pas β .

Comme pour la modélisation par offset, on peut représenter, pour différentes valeurs de β , les pourcentages d’allongements des trajectoires en fonction de l’anticipation t , en les tronquant lorsque la contrainte sur l’angle de déviation n’est pas respectée (voir figure 3.9).

FIGURE 3.9 – Allongements (en %) dûs au point tournant avec contraintes sur l'angle, l'avion dévié passant derrière (à gauche) ou devant (à droite), en fonction du temps d'anticipation t (l'unité de temps est le temps mis par les avions pour parcourir une séparation standard d). $\phi = \frac{\pi}{2}$.

FIGURE 3.10 – Augmentations des allongements (en %) dues au point tournant et aux différents offsets, l’avion dévié passant derrière (à gauche) ou devant (à droite), en fonction du temps d’anticipation t , $\phi = \frac{\pi}{2}$.

3.3 Comparaison offset - point tournant

On a représenté sur une même figure les augmentations de pourcentages d’allongements pour différents angles de mise en offset, ainsi que pour la modélisation point tournant (voir figure 3.10). L’angle d’incidence est de $\frac{\pi}{2}$. On observe que la modélisation point tournant est plus économique en terme d’allongement que la modélisation par offset, ceci quelle que soit l’anticipation choisie pour la résolution du conflit. On trouvera en annexe C les courbes représentant les pourcentages d’allongements des trajectoires avec nos deux modélisations pour différentes valeurs de l’angle maximal de déviation β (voir figures C.4 et C.5). Il apparaît que la résolution de conflits par mise en offset ne permet pas de résolution plus tardive, c’est à dire avec un temps d’anticipation plus faible.

On peut donc conclure que l’avantage majeur de la résolution par offset réside dans la linéarisation des contraintes d’évitement qui permet un calcul très rapide des valeurs de l’offset à utiliser. Néanmoins, la combinatoire reste la même dans les deux cas et un certain nombre de contraintes assez coûteuses doivent être vérifiées a posteriori lorsque l’on souhaite profiter de la linéarisation.

De plus, la linéarisation ne permet pas de prendre en compte les incertitudes sur les vitesses que nous aborderons dans la partie 3.6.

En conséquence, la linéarisation, intéressante sur le plan théorique, ne pourra vraisemblablement pas trouver d’application pratique.

FIGURE 3.11 – Rattrapages

3.4 Avions en rattrapage

Nous n'avons pas étudié jusqu'à présent les conflits causés par des rattrapages. Dans ce cas, lorsque deux avions sur la même route ont des vitesses proches, l'offset devient plus avantageux que le point tournant. En effet, que l'avion le plus rapide dépasse le plus lent ou que l'avion le plus lent évite le plus rapide (figure 3.11), l'offset permet d'éviter un éloignement de plus d'une séparation standard de la trajectoire initiale. Si l'on note V_x et V_y les vitesses des deux avions, r le rapport de ces vitesses, α la modification de cap de l'avion dévié, $a = \cos(\alpha)$, et Xd la demie longueur de la portion de trajectoire de l'avion dévié (d est la séparation standard), on peut alors montrer que si α est minimal alors :

$$(r - 1)^2 X^2 (1 - a^2) = (1 + r^2 - 2 r a)$$

La résolution de cette équation du second degré nous permet de calculer les valeurs $a(r, X)$ pour lesquelles α est minimal. On peut alors calculer le rapport des allongements avec point tournant et avec offset. L'allongement avec point tournant vaut $\frac{1-a}{a}$, l'allongement avec offset vaut $\frac{1-a}{Xa}$ et le rapport vaut donc tout simplement X .

Il est intéressant d'observer quelles valeurs doit prendre X en fonction de r pour que l'angle α reste raisonnable (inférieur à 30 degrés). La figure 3.12 donne les valeurs de X minimales pour que α reste inférieur à 30 degrés. On observe

FIGURE 3.12 – A gauche : X en fonction de r pour $r < 1$ - A droite : X en fonction de r pour $r > 1$.

que si les rapports de vitesses sont voisins de 1, l'intérêt de l'offset est clairement établi.

Il peut être également intéressant de savoir, lors d'un rattrapage, lequel des deux avions doit être dévié. On peut donc comparer à titre d'exemple les cas où :

- $r = 1.2, X = 10$.

- $r = \frac{1}{1.2}, X = \frac{10}{1.2}$

qui correspondent aux mêmes configurations mais en inversant les avions.

Dans le premier cas, $\alpha = 6.87$ degrés, ce qui génère un retard de $0.072 d$ pour le point tournant et 10 fois moins pour l'offset.

Dans le deuxième cas, $\alpha = 9.18$ degrés, ce qui génère un retard de $0.10 d$ pour le point tournant et 10 fois moins pour l'offset.

On en conclut que si l'on peut choisir l'angle de déviation, il vaut mieux modifier la trajectoire de l'avion le plus rapide.

Les calculs présentés ci-dessus nous conduisent finalement à la conclusion suivante : la modélisation point tournant, très efficace pour résoudre les problèmes de conflits à angle non nuls, s'avère insuffisante dans le cas des rattrapages. C'est pourquoi nous conserverons par la suite les deux modélisations, offset et point tournant.

3.5 Résolution en vitesse

La régulation en vitesse, aujourd'hui pratiquée principalement dans les phases de descente et d'approche, a l'avantage de ne pas changer le support de la trajectoire des avions. Néanmoins, elle nécessite des tenues de vitesses rigoureuses et des temps d'anticipation variables selon l'angle de conflit et sa gravité. Dans l'étude qui suit, nous supposons toujours que si les avions ne sont pas régulés, ils atteignent le point de croisement de leurs trajectoires au même instant. C'est le cas le plus contraignant. La première partie de cette étude analyse de façon théorique les temps d'anticipation nécessaires pour une régulation en vitesse, en supposant que les tenues de trajectoires $4D$ des avions sont parfaites. Les exemples pris autorisent des modulations de vitesses pouvant varier de 1 à 10 pour cent. La deuxième partie de l'étude reprend les résultats de la première partie en s'intéressant au cas des avions en descente.

3.5.1 Etude théorique pour des avions stables en croisière

Il s'agit d'étudier de façon théorique les conditions nécessaires pour une résolution de conflits en vitesse entre deux avions volant à altitude constante. Pour les avions 1 et 2, nous pouvons pour tout temps t définir leur position dans l'espace par :

$$\begin{aligned}\vec{x}_1(t) &= \vec{v}_1 t - \vec{r}_1 \\ \vec{x}_2(t) &= \vec{v}_2 t - \vec{r}_2\end{aligned}$$

Si l'on appelle d la séparation standard, la condition pour que les deux avions restent séparés pour tout t s'écrit :

$$\left(\frac{r_1}{v_1} - \frac{r_2}{v_2}\right)^2 (\|\vec{v}_1\|^2 \|\vec{v}_2\|^2 - (\vec{v}_1 \cdot \vec{v}_2)^2) \geq d^2 \|\vec{v}_1 - \vec{v}_2\|^2$$

Pour résoudre un conflit, en modifiant la vitesse de l'avion 1 par exemple, on remplace sa vitesse \vec{v}_1 par $(1+e)\vec{v}_1$. Si l'on appelle a l'angle formé par les trajectoires des deux avions, l'équation précédente s'écrit :

$$\begin{aligned}(1+e)^2 v_1^2 (d^2 - r_2^2 \sin(a)^2) \\ + 2(1+e) v_1 v_2 (\sin(a)^2 r_1 r_2 - d^2 \cos(a)) \\ + v_2^2 (d^2 - r_1^2 \sin(a)^2) \leq 0\end{aligned}$$

Posons :

$$\begin{aligned}v_1 &= v \\ v_2 &= r v \\ r_1 &= k d \\ r_2 &= r k d\end{aligned}$$

On se place alors dans l'hypothèse la plus pénalisante de conflit, les deux avions devant arriver au point de croisement des trajectoires au même instant.

L'équation précédente devient :

$$e^2 (1 - r^2 k^2 \sin(a)^2) + 2e(1 - r \cos(a)) + 1 + r^2 - 2r \cos(a) \leq 0$$

On peut résoudre cette équation en k ;

$$k = \frac{\sqrt{(1+e)^2 - 2\cos(a)r(1+e)} + r^2}{\sin(a)re}$$

Dans un premier temps, étudions le cas où les vitesses initiales des deux avions sont identiques.

Cas particulier : $v_1 = v_2$

On a alors :

$$k = \frac{\sqrt{(1+e)^2 - 2\cos(a)(1+e)} + 1}{\sin(a)e}$$

On peut alors représenter pour différentes valeurs de e les courbes donnant le temps d'anticipation nécessaire en fonction de l'angle des trajectoires. La figure 3.13 donne ces courbes pour $e = 1\%$, $e = 2\%$, $e = 3\%$, $e = 5\%$, $e = 10\%$ (du haut vers le bas). Pour l'exemple, on a choisit $v_1 = v_2 = 400 \text{ kt}$ et $d = 8 \text{ NM}$.

On constate immédiatement que :

- la résolution en vitesse est plus avantageuse pour des angles faibles.
- des modifications de vitesses inférieures à 5% nécessitent des temps d'anticipation de l'ordre d'une demi-heure.

Afin de préciser ce dernier point, on peut calculer l'angle "idéal" de résolution en vitesse en fonction de e . En effet, il suffit de chercher l'angle pour lequel l'anticipation est minimale, e étant fixé :

- si e est positif :

$$a = \arccos(1 + e)$$

- si e est négatif :

$$a = \arccos\left(\frac{1}{1+e}\right)$$

La courbe 3.14 donne les valeurs de cet angle en degrés pour e variant de -10 à 10% .

Le temps minimum d'anticipation pour l'angle "idéal" est donné par les formules suivantes :

- si e est positif :

$$t_{min} = \frac{(1+e)d}{ve}$$

FIGURE 3.13 – Temps d'anticipation en minutes en fonction de l'angle en degrés pour $e = 1\%$, $e = 2\%$, $e = 3\%$, $e = 5\%$ et $e = 10\%$ (de haut en bas), $v_1 = v_2 = 400 kt$, $d = 8 NM$.

FIGURE 3.14 – A gauche : angle minimum en degrés fonction de e en % - A droite : anticipation minimale en minutes en fonction de e en % (de haut en bas, $d = 8 NM$ et $d = 5 NM$), $v = 400 kt$.

– si e est négatif :

$$t_{min} = - \frac{d}{e v}$$

La courbe 3.14 donne le temps d'anticipation minimale en minutes pour cet angle en fonction de e en %, pour $v = 400 kt$, $d = 8 NM$ et $d = 5 NM$.

On observe ainsi que pour résoudre en vitesse des conflits avec un temps d'anticipation inférieur à 20 minutes, avec $v = 400 kt$ et $d = 8 NM$, il faut pouvoir modifier de plus de 5% la vitesse des avions, ce qui n'est pas toujours possible.

Cas général : $v_2 = r v_1$

Donnons tout d'abord deux exemples pour observer l'évolution des courbes précédentes. Dans le premier exemple, $e = 5\%$, $v_1 = 400 kt$, $d = 8 NM$ et l'on fait varier r (voir figure 3.15). On observe que plus l'avion non régulé est rapide, moins le temps d'anticipation est important. Néanmoins, il est intéressant de détailler la courbe précédente pour $r \geq 1$. Les courbes 3.15 donnent les temps d'anticipation minimaux pour $r = 1, 1.2, 1.4, 1.6, 1.8$ et 2 . On observe alors que les temps d'anticipation sont toujours supérieurs à 20 minutes et le minimum de chaque courbe dépend fortement de l'angle formé par les deux trajectoires. Ainsi, des vitesses très différentes déplacent l'angle "idéal".

On peut comme précédemment déterminer numériquement la valeur de "l'angle

FIGURE 3.15 – Anticipations minimales en minutes en fonction de l’angle d’incidence en degrés (de haut en bas, à gauche : $r = 0.5, 0.75, 1, 1.25, 1.5$ - à droite : $r = 1, 1.2, 1.4, 1.6, 1.8, 2$), $v_1 = 400 kt$, $d = 8 NM$.

idéal” en fonction de e et r ainsi que le temps d’anticipation minimum correspondant.

- si $1 + e \leq r$

$$a = \arccos\left(\frac{1+e}{r}\right) \quad t_{min} = \frac{d}{|e|v}$$

- si $1 + e \geq r$

$$a = \arccos\left(\frac{r}{1+e}\right) \quad t_{min} = \frac{(1+e)d}{v|e|r}$$

Les valeurs théoriques confirment l’observation précédente. Ainsi, un rapport de vitesses différent de 1 ne permet pas de réduire le temps d’anticipation minimum. Néanmoins, il déplace l’angle “idéal”. Lorsque deux avions se croisent avec des vitesses différentes, il se peut que le temps d’anticipation minimum soit plus faible que pour deux avions volant à vitesse identique (notamment si l’angle formé par les trajectoires est fort). Cependant, on retiendra que ce temps d’anticipation est toujours supérieur à $\frac{d}{ev}$. Le rapport $\frac{d}{ev}$ peut varier entre 1 et 2 minutes (si l’on prend $d = 8 NM$ et v variant de $500 kt$ à $250 kt$). Pour $e = 5\%$, le temps d’anticipation minimum varie donc de 20 à 40 minutes. Pour $e = 10\%$, il varie de 10 à 20 minutes. Il reste toutefois assez important.

3.5.2 Cas des avions en descente

Les avions en descente ont généralement une marge de manœuvre plus importante sur la régulation de leur vitesse. Ainsi, ils peuvent très souvent réduire leur

FIGURE 3.16 – Temps d’anticipation en minutes en fonction de e en % et de a en degrés. La courbe du bas est l’iso-temps valant 5 minutes. En allant vers l’extérieur, les iso-temps augmentent par pas de 1 minute ($r = 1$). La partie hachurée dépasse 20 minutes.

vitesse de plus de 10%. Aussi, dans cette partie, nous allons reprendre les résultats précédents avec e variant de -30% à 30% .

La figure 3.16 représente le temps d’anticipation minimum pour $r = 1$ (les deux avions ont la même vitesse) en fonction de e et de l’angle a des trajectoires ($v_1 = v_2 = 400 kt$ et $d = 8 NM$). Les courbes iso-temps parcourent les points pour lesquels le temps d’anticipation minimal est constant. La courbe intérieure correspond à un temps d’anticipation de 5 minutes. En allant de l’intérieur vers l’extérieur, le temps augmente par pas de 1 minute.

On observe ainsi que pour $e \geq 15\%$ et des angles de trajectoires entre 10 et 120 degrés, le temps d’anticipation reste inférieur à 10 minutes dans le pire des cas (c’est à dire lorsque les avions sont prévus au même instant au point de croisement des trajectoires). La résolution en vitesse peut alors s’appliquer de façon plus systématique.

La figure 3.17 reprend la figure 3.16 pour $r = 1.5$ et $r = 0.5$.

On observe que pour $r > 1$ (on intervient alors sur l’avion le plus lent) les résultats précédents restent bons. Si l’on intervient sur l’avion le plus rapide, les résultats se détériorent car il faut intervenir plus tôt.

Comme précédemment, on peut représenter (voir figure 3.18) la surface donnant l’angle “idéal” en fonction du rapport des vitesses r et de e .

FIGURE 3.17 – Temps d'anticipation en minutes en fonction de e en % et de a en degrés, à gauche : $r = 1.5$ - à droite : $r = 0.5$. La courbe du bas est l'iso-temps valant 5 minutes. En allant vers l'extérieur, les iso-temps augmentent par pas de 1 minute. La partie hachurée dépasse 20 minutes.

FIGURE 3.18 – En haut : angle “idéal” en fonction de r et e en %. - En bas : temps d’anticipation minimal en minutes en fonction de r et e en %. Les courbes iso-temps augmentent par pas de 5 minutes en allant de l’extérieur vers l’intérieur. La courbe iso-temps extérieure correspond aux points pour lesquels le temps vaut 5 minutes.

A chaque angle “idéal” correspond un temps d’anticipation minimal que nous avons représenté figure 3.18. Les courbes iso-temps augmentent par pas de 5 minutes en allant de l’extérieur vers l’intérieur. La courbe iso-temps extérieure correspond aux points pour lesquels le temps vaut 5 minutes.

L’observation des courbes précédentes montre qu’une régulation en vitesse est envisageable pour une majorité de conflits dans la mesure où la marge de régulation sur les avions est suffisamment importante. Ainsi, si l’on peut faire varier la vitesse des avions de plus de 15%, on peut espérer résoudre en vitesse bon nombre de conflits.

Nous n’avons pas, jusqu’à présent, évoqué les problèmes d’incertitude sur les tenues de vitesses des avions. Si les avions ne peuvent tenir leurs vitesses de façon précise, la régulation en vitesse devient difficile. Dans les hypothèses que nous avons choisies, nous avons toujours considéré les conflits dans les cas les plus contraignants, à savoir lorsque les deux avions sont supposés passer par le point d’intersection de leurs trajectoires au même instant. Si deux avions sont déjà partiellement séparés, une régulation en vitesse peut permettre de les séparer plus facilement. Néanmoins, plus on anticipe la régulation, plus l’incertitude joue un rôle important. Ainsi, à 400 *kt*, une incertitude de 1% pendant 20 minutes représente 1.33 *NM*. Aujourd’hui, peu d’avions sont équipés de *FMS – 4D* permettant de telles précisions. Ceci explique pourquoi les contrôleurs n’utilisent la régulation en vitesse qu’en phase de descente, les marges de manœuvre étant alors plus importantes.

Remarque

Cette étude a permis d’expliquer pourquoi actuellement la régulation en vitesse intervient principalement dans les phases de descente des avions. Compte tenu des faibles marges dont on dispose dans le plan horizontal, nous abandonnerons les possibilités de régulation en vitesses dans la modélisation.

FIGURE 3.19 – Modélisation de l'incertitude.

3.6 Modélisation de l'incertitude.

3.6.1 Modélisation

Si un avion est capable de suivre précisément une route et une altitude, maîtriser précisément sa vitesse sol n'est pas possible en raison notamment des vents. Par ailleurs, les réacteurs ont un régime de consommation idéal, ne doivent pas être soumis à des changements de régime permanents, et donc, ne peuvent pas réajuster constamment la vitesse. On modélisera donc les avions à l'instant t dans le plan horizontal, non pas par des points, mais par des segments de droite. Il suffira alors de vérifier, pour que la contrainte horizontale soit respectée, que les segments de droite qui modélisent les avions sont séparés par la séparation standard (voir figure 3.19). Dans le plan vertical, l'incertitude est beaucoup plus forte. Le même principe que dans le plan horizontal est repris mais les pourcentages d'incertitudes choisis seront beaucoup plus forts.

3.6.2 Evaluation de la probabilité de conflit

Compte tenu de l'incertitude sur la vitesse des avions que nous observons, il est évident qu'un conflit ne peut être prévu à l'avance avec certitude. Il se peut en effet

que les erreurs sur la vitesse de chacun des avions suffisent à éviter le conflit. Dans ce cas, une manœuvre d'évitement s'avère inopportune. Le but de ce paragraphe est de mesurer la valeur de la probabilité de conflit en fonction de la configuration du conflit et de l'incertitude sur les vitesses.

Configurations étudiées

On ne peut bien évidemment pas étudier toutes les configurations de conflits. Aussi, nous limiterons au cas où les deux avions ont la même vitesse V et convergent sous un angle ϕ vers un point de conflit. Les avions sont distants du point de conflit de VT . On suppose que chaque avion se déplace à vitesse constante, avec une incertitude en pourcentage sur cette vitesse notée Er . Pour simplifier, on supposera donc que chacun des deux avions a une probabilité uniforme d'avoir sa vitesse comprise entre $(1 - Er)V$ et $(1 + Er)V$. On notera d la norme de séparation choisie. On supposera que si les avions respectent précisément leurs vitesses, ils atteignent le point de croisement de leurs trajectoires au même instant. On veut mesurer la probabilité de conflit en fonction de l'anticipation T , de l'angle d'incidence ϕ et de l'incertitude Er choisie.

Calcul mathématique

On numérotera 1 et 2 les indices des deux avions. A un instant $t \geq -T$, les avions sont repérés par leur position dans le plan :

$$\begin{aligned} & -T \vec{V}_1 + t(1 + Er_1) \vec{V}_1 \\ & -T \vec{V}_2 + t(1 + Er_2) \vec{V}_2 \end{aligned}$$

avec Er_1 et Er_2 compris entre $-Er$ et Er . A partir des positions des deux avions à tout instant t , on cherche à quelle condition sur Er_1 et Er_2 il y a conflit. Pour cela, on est amené à déterminer le signe du discriminant d'une équation du second degré. La condition, une fois simplifiée, s'écrit de la façon suivante.

Il y a conflit si et seulement si :

$$\frac{\frac{V^2 T^2}{d^2} (\sin \phi)^2 - 1}{2(1 - \cos \phi)} \leq \frac{(1 + Er_1)(1 + Er_2)}{(Er_1 - Er_2)^2}$$

Il est clair que si l'angle ϕ vaut 0 ou π , le conflit est certain. On remarquera que la condition précédente ne dépend que du rapport $x = \frac{VT}{d}$ qui mesure l'anticipation par rapport au point de conflit. On peut représenter la condition par une fonction $f(x, \phi, Er_1, Er_2)$ qui vaut 0 lorsque la condition n'est pas satisfaite et 1 lorsque la condition est satisfaite. On intègre ensuite cette fonction sur le pavé des erreurs possibles et on obtient la probabilité $p(x, \phi, Er)$ pour que les deux avions entrent en conflit avec une anticipation x , un angle d'incidence ϕ et une erreur sur les vitesses Er .

Résultats numériques

A titre d'illustration, nous allons observer l'évolution de la probabilité de conflit lorsque :

- l'anticipation $x = \frac{VT}{d}$ augmente (voir figure 3.20), on fixera $\phi = \frac{\pi}{2}$, $Er = 5\%$, $V = 400 kt$, $d = 10 NM$ et on fera varier T de 0 à 120 minutes. On constate que la probabilité de conflit vaut 1 jusqu'à 20 minutes, après quoi elle décroît. On remarquera qu'elle décroît alors de moins en moins vite avec le temps.
- l'angle ϕ varie de 0 à π (voir figure 3.20), on fixera $V = 400 kt$, $d = 10 NM$, $T = 30$ minutes, et $Er = 5\%$. On observe que l'angle ϕ a une très forte influence sur la probabilité de conflit. En effet, dans cet exemple, celle-ci varie de 0.75 à 1 selon que les avions convergent à 20 degrés ou 120 degrés. On observe donc que le risque de conflit est d'autant plus faible que l'angle d'incidence est faible. Or, on sait que le conflit est d'autant plus difficile à résoudre que cet angle est faible. On en déduit que les conflits d'avions convergents sont à la fois moins certains et plus difficiles à résoudre. Il convient de mettre un bémol à cette affirmation. En réalité, l'incertitude sur les vitesses n'a pas une distribution uniforme et dépend notamment, lorsqu'elle est provoquée par le vent, du cap de chaque avion. Deux avions ayant des caps voisins auront une forte probabilité d'être influencés dans le même sens par le vent. L'incertitude sur leur vitesse relative est alors plus faible. Nous ne tenons toutefois pas compte de ce phénomène dans cette étude.
- le pourcentage d'erreur Er varie de 0 à 20% (voir figure 3.21), on fixera $V = 400 kt$, $d = 10 NM$, $T = 30$ minutes, et $\phi = \frac{\pi}{2}$. La sensibilité de la probabilité de conflit au pourcentage d'erreur est semblable à sa sensibilité au temps d'anticipation.
- l'angle ϕ varie de 0 à π et le temps d'anticipation varie de 0 à 120 minutes (voir figure 3.21), on fixera $V = 400 kt$, $d = 10 NM$ et $Er = 5\%$. On visualise alors bien le lien entre la probabilité de conflit et l'angle d'incidence. Le choix du moment de résolution est fortement dépendant de la géométrie du conflit.

Il apparaît donc au travers de ces résultats, qu'une fois de plus, les conflits à faible incidence sont ceux qui posent le plus de problèmes. Moins "certains", ils sont également plus difficiles à résoudre.

3.6.3 Espérance de coût de résolution de conflit

Le coût de résolution du conflit

Si l'on veut résoudre un conflit de façon certaine avec un temps d'anticipation T , il faut intégrer l'incertitude sur les vitesses dans la résolution. Ainsi, au lieu de séparer des points par une séparation standard, il faut séparer des segments de droite dont la longueur augmente avec le temps. Pour cela, on peut mesurer l'al-

FIGURE 3.20 – Probabilité de conflit en fonction (à gauche, pour $\phi = \frac{\pi}{2}$) du temps d'anticipation en minutes, (à droite, pour $T = 30$ minutes), de l'angle d'incidence en degrés, $Er = 5\%$, $V = 400 kt$, $d = 10 NM$.

FIGURE 3.21 – Probabilité de conflit en fonction (à gauche, pour $\phi = \frac{\pi}{2}$ et $T = 30$ minutes) du pourcentage d'erreur sur les vitesses, (à droite pour $Er = 5\%$) de l'angle d'incidence en degrés et du temps d'anticipation en minutes, $V = 400 kt$, $d = 10 NM$.

FIGURE 3.22 – Allongement de la trajectoire en nautiques en fonction du temps d’anticipation en minutes, modélisation par offset (à gauche), modélisation point tournant (à droite), $\phi = \frac{\pi}{2}$, $Er = 5\%$, $d = 10 \text{ NM}$.

longement de la trajectoire en nautiques lorsque l’on dévie un des deux avions, en fonction du temps d’anticipation engendré par la résolution d’un conflit. Pour ce calcul, nous garderons les mêmes caractéristiques que précédemment pour les deux avions. Nous choisirons un angle d’incidence de $\frac{\pi}{2}$ et une erreur sur la vitesse de 5%. Le calcul est fait dans les deux modélisations offset et point tournant (voir figure 3.22). Pour la modélisation par offset, le choix de l’angle de déviation a été fixé à 30 degrés de façon à ce que la résolution se fasse toujours pendant la phase d’offset (Un contrôleur humain proposant un offset donnera généralement un angle de déviation de 10, 20 ou 30 degrés). De plus, afin de préserver la linéarité du problème dans la phase d’offset, les avions ne seront pas modélisés par des segments de droite mais des sphères dont la taille augmente en fonction de l’incertitude avec le temps. En conséquence, les allongements de trajectoires obtenus figure 3.22 sont très importants comparés à ceux de la modélisation par point tournant, pour laquelle l’avion est dévié le plus tôt possible et l’angle de déviation n’est pas contraint. Les résultats obtenus pour les deux modélisations ne peuvent donc pas être comparés. Nous souhaitons comparer, dans le cas présent, la forme des deux courbes.

On observe qu’avec la modélisation par offset, l’évitement le plus tardif est le plus économique. La courbe est en fait une droite. L’allongement est donc une fonction linéaire du temps d’anticipation. Pour la modélisation point tournant, au phénomène d’accroissement de l’allongement provoqué par l’incertitude s’ajoute un phénomène inverse. Plus l’on anticipe le conflit, plus la déviation de l’avion dévié sera faible.

FIGURE 3.23 – Espérance du coût de résolution en fonction du temps d'anticipation, modélisation par offset (à gauche), modélisation point tournant (à droite), $\phi = \frac{\pi}{2}$, $Er = 5\%$, $d = 10 NM$.

L'espérance du coût de résolution de conflit

On souhaiterait déterminer le moment optimal pour résoudre un conflit à deux avions en tenant compte de ce phénomène d'incertitude. Pour cela, il faut préciser quel est le critère d'optimalité retenu. N'ayant pas de certitude sur l'existence ou non d'un conflit lorsque le temps d'anticipation augmente, on ne peut pas parler de coût de résolution. Par contre on peut définir l'espérance du coût de résolution de conflit. Pour tout temps d'anticipation T , on connaît la probabilité pour que le conflit se produise et le coût associé à ce temps. Il suffit de multiplier les deux pour obtenir l'espérance du coût de résolution du conflit sachant que s'il n'y a pas de conflit, le coût de résolution est nul. On observera la figure 3.23 pour les deux modélisations offset et point tournant. Si pour l'offset, l'espérance de coût de résolution est minimale pour un temps d'anticipation le plus faible possible, pour la modélisation point tournant, dans cet exemple, le coût de résolution est minimal pour une anticipation de 20 minutes ou 130 nautiques.

Les calculs que nous venons de présenter permettent de conclure qu'une résolution par offset doit être la plus tardive possible. Pour une résolution par point tournant, il existe un compromis qui minimise l'espérance de coût de résolution du conflit. Ce compromis dépend de l'incertitude sur les vitesses et de la configuration du conflit.

Cette étude n'est pas exhaustive mais propose un critère d'évaluation de l'instant optimal de début de résolution de conflit.

3.7 Modélisation temps réel

Les études précédentes ont conduit aux conclusions suivantes :

- On peut modéliser une manœuvre d'évitement horizontale entre deux avions par un point tournant.
- La modélisation par offset reste indispensable pour les cas de rattrapage.
- L'incertitude sur les vitesses des avions et sur leurs taux de montée doit être prise en compte et fait apparaître, dans le cas de la modélisation point tournant, un temps "idéal" de début de manœuvre.

D'autres contraintes pratiques sont maintenant introduites dans le modèle dans un but opérationnel :

- Les angles de déviation pour le point tournant et l'offset seront discrétisés et pourront donc prendre pour valeur : -30 , -20 , -10 , 0 , 10 , 20 et 30 degrés (ceci afin de pouvoir donner aux pilotes des ordres simples).
- Une manœuvre commencée ne pourra pas être remise en cause, on pourra néanmoins modifier son ampleur en veillant à ce qu'elle reste point tournant ou offset.
- Le système ne proposera pas de régulation de vitesse.
- Les manœuvres d'évitement proposées sont horizontales. Le système ne pourra donc pas modifier la trajectoire verticale d'un avion même s'il peut faire tourner des avions évolutifs (en montée ou en descente). Une telle hypothèse est relativement restrictive : un certain nombre de résolutions ne seront plus optimales, en particulier, lorsqu'un des avions impliqués dans un conflit est évolutif. Les contrôleurs emploient alors généralement une manœuvre dans le plan vertical (mise en palier) pour les avions évolutifs (en montée ou en descente). Ces manœuvres sont souvent meilleures qu'une déviation horizontale lorsque les avions sont sur une route identique ou opposée. La dimension verticale devra nécessairement être intégrée dans les évolutions futures. Néanmoins l'ajout de manœuvres dans le plan vertical ne complexifie pas de façon significative le modèle.

Nous obtenons donc le modèle suivant (figure 3.24) où une manœuvre est déterminée par les 4 paramètres suivants :

- la date de début du virage d'éloignement t_0
- la date de fin du virage d'éloignement t_1
- la date de début du virage de retour sur la trajectoire initiale t_2
- l'angle de déviation de la manœuvre : α

Les différents t_i représentent des dates relatives. L'instant $t = 0$ correspond au début de la prévision. A cette date, la position de l'avion est connue sans incertitude¹. On la considère comme *la position initiale* de l'avion.

Dans notre modélisation, le virage de retour a le même angle α mais dans le sens opposé. La durée de la phase de retour est donc égale à la durée de la phase d'éloignement. La date de fin de manœuvre, notée t_3 n'est donc pas un paramètre

1. L'incertitude "Radar" est prise en compte par la séparation standard

FIGURE 3.24 – Modèles de manœuvre.

de la manœuvre et on a : $t_3 = t_2 + t_1 - t_0$

On aurait pu choisir un angle de retour différent de l'angle de déviation initial. Dans la pratique, les pilotes essaient de négocier une route directe sur la prochaine balise ou la destination finale. Modifier le modèle en ce sens ne devrait pas poser de problème mais nous avons souhaité dans un premier temps simplifier la modélisation.

Si l'on est en présence d'un point tournant, on aura $t_2 = t_1$.

Un tel modèle réduit donc la taille du problème de façon significative. Pour un conflit impliquant n avions, la dimension de l'espace de recherche sera de $4n$. Ceci nous permettra de résoudre des gros conflits sans avoir à explorer un domaine de solutions admissibles trop vaste.

3.7.1 Modèle de cluster

Nous nous proposons de résoudre non pas uniquement des conflits élémentaires mais des situations plus complexes. Nous avons donc introduit la notion de *clusters*. Un cluster est le résultat d'une fermeture transitive dans l'ensemble des conflits sur un horizon temporel fixé, que nous appellerons fenêtre d'anticipation F_a . Si un avion A est en conflit avec B à l'instant t_a et B est en conflit avec C à l'instant t_c (avec $t_a < F_a$ et $t_c < F_a$) alors A , B et C appartiennent au même cluster. Ainsi dans la figure 3.25, les avions A et B ne sont jamais en conflit mais ils

FIGURE 3.25 – Exemple de cluster.

appartiennent au même cluster. Une déviation de C pour éviter B peut permettre également à C d'éviter A . La résolution des conflits du clusters doit donc être globale.

3.7.2 Déroulement en temps réel

Un des objectifs retenus est de pouvoir détecter et résoudre les conflits en temps réel. Fixons un délai Δ . La situation doit être optimisée toutes les Δ secondes à partir de nouvelles données. Le système dispose donc d'un temps inférieur à Δ pour analyser la nouvelle situation et trouver les trajectoires optimales sans conflit pour l'ensemble des avions impliqués dans la fenêtre d'anticipation F_a . Le reste de l'intervalle Δ est consacré à la transmission des manœuvres du sol vers le bord. Durant l'intervalle Δ , les avions volent toujours. Le système ne peut donc changer la trajectoire prévue de chacun des avions pendant les Δ premières secondes de F_a . Ces trajectoires correspondent donc à la prévision effectuée au tour précédent. Par contre le système reste libre de modifier toutes les trajectoires entre Δ et F_a à l'exception des branches de retour en point tournant ou en offset. En effet, cette partie de la manœuvre est complètement déterminée par le début de la manœuvre.

Appelons t le temps relatif utilisé par le système de résolution, T l'échelle de temps absolu et T_0 la date de début de la première itération de notre processus de résolution. A chaque itération i , le processus analyse une nouvelle situation entre $t = 0$ et $t = F_a$. La situation prévue à l'instant $t = 0$ correspond à la situation réelle existant à $T = T_0 + i\Delta$ en considérant que $i = 0$ pour la première itération. La figure 3.26 peut alors se commenter de la façon suivante :

FIGURE 3.26 – Séquencement en temps réel.

itération 1 ($T = T_0$) : Le système prévoit un offset pour $t > 2\Delta$.

itération 2 ($T = T_0 + \Delta$) : Le système a réduit l'offset en un point tournant prévu pour $t > \Delta$.

itération 3 ($T = T_0 + 2\Delta$) : Le système a réduit le point tournant prévu. La date de début de manœuvre est prévue pour $t < \Delta$. Ce début de manœuvre sera donc communiqué au pilote. Par rapport au modèle présenté, on a fixé les variables t_0 et α .

itération 4 ($T = T_0 + 3\Delta$) : L'avion a effectué son début de manœuvre et le système prévoit toujours le même point tournant. Le reste de la manœuvre est donc à son tour fixé : t_1 et t_2 (en l'occurrence $t_1 = t_2$ car il s'agit d'un point tournant).

itération 5 ($t = T_0 + 4\Delta$) : L'avion est sur sa branche de retour.

Conclusion

La modélisation simplifiée par point tournant ou offset permet dans le cas de conflits à deux avions d'approcher avec une perte acceptable les trajectoires optimales. Dans la mesure où il paraît peu envisageable de donner à un avion plus d'un ordre de manœuvre à la fois, cette modélisation pourra être conservée pour la résolution de conflits à n avions avec $n > 2$. L'intérêt principal d'une telle modélisation est qu'elle permet de réduire fortement la taille de l'espace de recherche. On peut désormais définir une trajectoire d'évitement par la donnée d'un plus trois temps et d'un angle de déviation.

On a pu également observer dans ce chapitre que résoudre un conflit en modifiant seulement la vitesse des avions requiert une prise de décision fortement anticipée compte tenu des faibles marges de manœuvre des avions (en route) et des incertitudes sur leurs tenues de vitesses. Une modification de vitesse pourrait donc être efficace à condition de pouvoir la combiner avec des manœuvres de type point tournant ou offset, hypothèse que nous avons pour l'instant écartée.

Afin de pouvoir rendre le modèle réaliste, il est essentiel de tenir compte de l'incertitude sur les vitesses, taux de montée et de descente dans le modèle. La notion de séparation standard est ainsi remplacée par des volumes d'occupation de l'espace, se déformant avec le temps, représentant les positions potentielles futures des avions.

Enfin, nous n'avons pas souhaité, dans un premier temps, introduire de manœuvre dans le plan vertical, mais la modélisation devra, dans une prochaine étape, prendre en compte ce type de manœuvre.

Chapitre 4

Résolution numérique par Algorithmes Génétiques

Introduction

Nous avons pu mettre en évidence l'inefficacité d'une méthode locale pour la résolution de conflits dans la partie 2.4. Parmi les méthodes d'optimisation globale, les algorithmes génétiques ont la particularité de ne requérir que peu d'information sur la fonction d'évaluation et de pouvoir détecter simultanément plusieurs optima locaux proches de l'optimum. C'est principalement ce dernier point qui a motivé le choix de tels algorithmes pour notre problème. En effet, en vue de créer un outil d'aide à la décision, il peut être intéressant de rechercher non pas la solution optimale mais un ensemble de solutions différentes proches de l'optimum.

De plus, les AGs sont relativement robustes et ne requièrent *a priori* aucune hypothèse de régularité sur la fonction à optimiser (leur efficacité dépend cependant fortement de la régularité de cette dernière). La fonction d'évaluation que l'on veut optimiser peut être le résultat d'une simulation, ce qui sera le cas pour le problème de résolution de conflits.

Parmi les autres méthodes globales :

- les méthodes de recuit simulé ne garantissent pas plus que les algorithmes génétiques l'optimalité de la solution trouvée, ils ont l'inconvénient de ne proposer qu'une seule solution. Par ailleurs, ils n'opèrent que des mutations qui ne tiennent pas compte de la structure du problème, en particulier de la séparabilité partielle que nous aborderons dans la partie 4.5.
- les méthodes de type programmation par intervalles [Han92, RR95] mériteraient sans doute d'être envisagées. On peut néanmoins craindre que la structure de la fonction d'évaluation rende très coûteuse en temps de calcul sa mise en œuvre.
- d'autres méthodes globales existent [HP95]. Elles requièrent généralement des propriétés particulières que ne vérifie pas notre fonction d'évaluation (concavité, fonction lipschitzienne...).

Pour toutes ces raisons, nous avons fait le choix d'utiliser, dans ces travaux, les algorithmes génétiques.

Les premiers travaux sur les algorithmes génétiques ont commencé dans les années cinquante lorsque plusieurs biologistes américains ont simulé des structures biologiques sur ordinateur. Puis entre 1960 et 1970, John Holland [Hol75], sur la base des travaux précédents, développa les principes fondamentaux des algorithmes génétiques dans le cadre de l'optimisation mathématique. Malheureusement, les ordinateurs de l'époque n'étaient pas assez puissants pour envisager l'utilisation des algorithmes génétiques sur des problèmes réels de grande taille. L'ouvrage de Goldberg [Gol89c] qui décrit l'utilisation des algorithmes génétiques dans le cadre de la résolution de problèmes concrets a permis de mieux faire connaître ces derniers et a marqué le début d'un nouvel intérêt pour cette technique d'optimisation.

Sur le plan théorique, les résultats sont apparus très tardivement, probablement à cause de la complexité théorique induite par ces algorithmes. Des résultats théoriques de convergence stochastique pourront être trouvés dans [vLA87, Zhi91]. En 1993 une démonstration complète de convergence stochastique est établie par R.Cerf [Cer94, FW83]. Néanmoins, ces résultats théoriques, sur lesquels nous reviendrons par la suite sont difficilement exploitables dans la pratique. La performance pratique des algorithmes génétiques fait beaucoup appel au savoir faire de l'utilisateur.

Les principes généraux des algorithmes génétiques seront rappelés dans la première partie. Les principaux résultats théoriques sont abordés partie deux. La partie trois donne une description détaillée des AGs utilisés. Les améliorations classiques apportées sont abordées partie quatre. La partie cinq s'intéresse à la création d'opérateurs génétiques adaptés aux fonctions partiellement séparables. Des exemples d'application sur des fonctions test classiques sont alors proposés et discutés. La partie six est consacrée à l'adaptation de ces opérateurs aux problèmes de conflits. De nombreux exemples pratiques sont résolus et la qualité des solutions obtenues est évaluée sur un conflit à 5 avions.

4.1 Principes généraux

Les algorithmes génétiques tentent de simuler le processus d'évolution naturelle dans un environnement donné. Ils utilisent un vocabulaire similaire à celui de la génétique naturelle. Cependant, les processus naturels auxquels ils font référence sont beaucoup plus complexes. On parlera ainsi d'individu dans une population. L'individu est composé de chromosomes. Les chromosomes sont eux-mêmes constitués de gènes qui contiennent les caractères héréditaires de l'individu. Les principes de *sélection*, de *croisement*, de *mutation* s'appuient sur les processus naturels du même nom.

Pour un problème d'optimisation donné, un individu représente un point de l'espace d'états. On lui associe la valeur du critère à optimiser. On génère ensuite de

FIGURE 4.1 – Principe général des algorithmes génétiques

façon itérative des populations d'individus sur lesquelles on applique des processus de sélection, de croisement et de mutation. La sélection a pour but de favoriser les meilleurs éléments de la population, le croisement et la mutation assurent une exploration efficace de l'espace d'états.

Le principe général des algorithmes génétiques est décrit par la figure 4.1.

On commence par générer une population d'individus. Pour passer d'une génération k à la génération $k + 1$, les trois opérations suivantes sont répétées pour tous les éléments de la population k . Des couples de parents P_1 et P_2 sont sélectionnés en fonction de leurs adaptations. L'opérateur de croisement leur est appliqué avec une probabilité P_c (généralement autour de 0.6) et génère des couples d'enfants C_1 et C_2 . D'autres éléments P sont sélectionnés en fonction de leur adaptation. L'opérateur de mutation leur est appliqué avec la probabilité P_m (P_m est généralement très inférieur à P_c) et génère des individus mutés P' . Le niveau d'adaptation des enfants (C_1, C_2) et des individus mutés P' sont ensuite évalués avant insertion dans la nouvelle population.

Différents critères d'arrêt de l'algorithme peuvent être choisis :

- Le nombre de générations que l'on souhaite exécuter peut être fixé *a priori*. C'est ce que l'on est tenté de faire lorsque l'on doit trouver une solution dans un temps déterminé.

- L’algorithme peut être arrêté lorsque la population n’évolue plus ou plus suffisamment rapidement.

Pour utiliser un algorithme génétique sur un problème particulier, on doit donc disposer des cinq éléments suivants :

1. Un principe de codage de l’élément de population. Cette étape associe une structure de données à chacun des points de l’espace d’états. Elle se place généralement après une phase de modélisation mathématique du problème traité. La qualité du codage des données conditionne le succès des algorithmes génétiques. Les codages binaires ont été très utilisés à l’origine. Les codages réels sont désormais largement employés, notamment dans les domaines applicatifs pour l’optimisation de problèmes à variables réelles.
2. Un mécanisme de génération de la population initiale. Ce mécanisme doit être capable de produire une population d’individus non homogène qui servira de base pour les générations futures. Le choix de la population initiale est important car il peut rendre plus ou moins rapide la convergence vers l’optimum global. Dans le cas où l’on ne connaît rien du problème à résoudre, il est essentiel que la population initiale soit répartie sur tout le domaine de recherche.
3. Une fonction à optimiser. Celle-ci retourne une valeur de \mathbb{R}^+ appelée *fitness* ou fonction d’évaluation de l’individu.
4. Des opérateurs permettant de diversifier la population au cours des générations et d’explorer l’espace d’états. L’opérateur de croisement recompose les gènes d’individus existant dans la population, l’opérateur de mutation a pour but de garantir l’exploration de l’espace d’états.
5. Des paramètres de dimensionnement : taille de la population, nombre total de générations ou critère d’arrêt, probabilités d’application des opérateurs de croisement et de mutation.

Avant de détailler chacun de ces points, rappelons brièvement les principaux travaux théoriques existants sur les algorithmes génétiques.

4.2 Résultats théoriques

Les algorithmes génétiques ont montré leur efficacité pratique bien avant que les résultats de convergence théorique soient établis. Aujourd’hui, nous disposons de trois approches théoriques différentes permettant de mieux comprendre le fonctionnement asymptotique de ces algorithmes.

- La théorie des schémas [Hol75, Gol89a] constitue une première approche du problème. S’appliquant à des chaînes de bits, elle étudie le comportement asymptotique de l’algorithme et l’effet des différents opérateurs sur la structure des schémas. Très simplifiée, cette théorie donne une bonne approche intuitive des problèmes que posent les algorithmes génétiques. Quelques résultats de la théorie des schémas sont présentés en annexe E.1.

- La deuxième approche découle des résultats de convergence stochastique sur des méthodes de recuit simulé [vLA87, Zhi91]. Sous certaines hypothèses, on montre qu'en un temps infini, l'algorithme génétique converge grâce à l'opérateur de mutation. Le rôle de l'opérateur de croisement reste ignoré, ce qui n'est pas satisfaisant. En effet, dans la pratique, l'opérateur de croisement est tout à fait déterminant.
- L'approche théorique la plus récente est proposée dans les travaux de Raphael Cerf [Cer94], qui utilise la théorie de Freidlin et Wentzell [FW83] pour étudier les états stables de l'algorithme génétique, pris comme un processus Markovien. On trouvera un résumé de ces résultats en annexe E.2. Le troisième article proposé par Cerf met notamment en évidence l'intérêt de l'opérateur de croisement au travers d'un algorithme génétique utilisant une méthode de sélection particulière assurant la diversité de la population.

4.3 Description détaillée

Pour l'implémentation pratique, on se ramènera toujours au cas d'une fonction à maximiser comprise entre 0 et 1.

4.3.1 Codage des données

Historiquement le codage utilisé par les algorithmes génétiques était représenté sous forme de chaînes de bits contenant toute l'information nécessaire à la description d'un point dans l'espace d'états. Ce type de codage a pour intérêt de permettre la création d'opérateurs de croisement et de mutation simples. C'est également en utilisant ce type de codage que les premiers résultats de convergence théorique ont été obtenus.

Cependant, ce type de codage n'est pas toujours bon. En effet, deux éléments voisins en terme de distance de Hamming¹ ne codent pas nécessairement deux éléments proches dans l'espace de recherche. Cet inconvénient peut être évité en utilisant un codage de Gray. Toutefois, pour des problèmes d'optimisation dans des espaces de grande dimension, le codage binaire peut rapidement devenir mauvais. Généralement, chaque variable est représentée par une partie de la chaîne de bits et la structure du problème n'est pas bien reflétée, l'ordre des variables ayant une importance dans la structure du chromosome alors qu'il n'en a pas forcément dans la structure du problème.

Les algorithmes génétiques utilisant des vecteurs réels [Gol91, Wri91] évitent ce problème en conservant les variables du problème dans le codage de l'élément de population sans passer par le codage binaire intermédiaire. La structure du problème est conservée dans le codage. Ce type de codage sera utilisé pour le problème de résolution de conflits.

1. Pour plus d'information sur les distances de Hamming et codages de Gray, voir [Gol89c]

Dans chacune des applications suivantes, le codage des données sera précisé ainsi que l'espace d'états sur lequel on travaille.

4.3.2 L'algorithme génétique élémentaire

On définit les variables globales suivantes :

- N : nombre total de générations
- K : nombre d'éléments de population
- P_c : pourcentage de croisements
- P_m : pourcentage de mutations

On définit par ailleurs deux listes POP et $NPOP$ contenant K éléments.

Chaque élément se compose de deux champs :

- un champs ($POP.data$) de la taille d'un individu de l'espace d'états.
- un champs ($POP.fitness$) qui permettra de conserver sa fitness.

L'algorithme génétique élémentaire a la forme suivante :

- $INITIALISER[POP]$
- Répéter pour k variant de 0 à N :
 1. $EVALUER[POP]$
 2. $SELECTION[NPOP, POP]$
 3. $CROISEMENT[NPOP]$
 4. $MUTATION[NPOP]$
 5. $POP = NPOP$
- $EVALUER[POP]$
- $MEILLEUR[POP]^2$

4.3.3 Génération de la population initiale : $INITIALISER[POP]$

Le choix de la population initiale d'individus peut conditionner fortement la rapidité de l'algorithme. Si la position de l'optimum dans l'espace d'états est totalement inconnue, il est naturel de générer aléatoirement des individus en faisant des tirages uniformes dans chacun des domaines associés aux composantes de l'espace d'états, en veillant à ce que les individus produits respectent les contraintes [MJ91]. Si par contre, des informations *a priori* sur le problème sont disponibles, il paraît naturel de générer les individus dans un sous-domaine particulier afin d'accélérer la convergence. Dans l'hypothèse où la gestion des contraintes ne peut se faire directement, les contraintes sont généralement incluses dans le critère à optimiser sous forme de pénalités. Cependant, lorsque c'est possible, ne générer que des éléments de population respectant les contraintes permet de limiter la taille de l'ensemble sur lequel on optimise la fonction.

- Pour le problème de résolution de conflits, la satisfaction des contraintes ne peut se faire *a priori*. Nous détaillerons partie 4.6.3 comment nous générons la population initiale.

2. La procédure $MEILLEUR[POP]$ retourne le meilleur élément de la population.

- Pour les autres applications, la population initiale sera générée aléatoirement avec une distribution uniforme sur tout l'espace d'états.

4.3.4 Evaluer la population : $EVALUER[POP]$

Il s'agit de considérer un par un les éléments de population et de les évaluer. Si F est la fonction à maximiser :

$EVALUER[POP]$

- Pour i variant de 1 à K
 $POP(i).fitness = F[POP(i).data]$

Gestion des contraintes

Un élément de population qui viole une contrainte se verra attribuer une mauvaise fitness et aura une probabilité forte d'être éliminé par le processus de sélection.

Dans le problème de résolution de conflits, il peut cependant être intéressant de conserver, tout en les pénalisant, les éléments non admissibles car ils peuvent permettre de générer des éléments admissibles de bonne qualité. Pour s'en convaincre, on pourra observer la position des optima dans \mathbb{R}^4 pour le problème de conflit à deux avions utilisant la modélisation point tournant (voir figureD.8) lorsque l'on fixe une des deux trajectoires. Dans ce cas, si l'on se contente d'affecter la valeur 0 à une trajectoire non admissible, on peut trouver des configurations pour lesquelles l'optimum ne peut être trouvé que par hasard. Plus généralement, pour le problème de résolution de conflits, l'optimum est atteint lorsque l'une au moins des contraintes de séparation est saturée. On se retrouve donc toujours à l'optimum sur la frontière de l'espace admissible.

Gérer les contraintes en pénalisant la fonction fitness est difficile, un "dosage" s'impose pour ne pas favoriser la recherche de solutions admissibles au détriment de la recherche de l'optimum ou inversement. De plus, dans un but opérationnel (pour des raisons évidentes de sécurité), il paraît essentiel qu'un élément de population admissible ait une fitness supérieure à un élément ne respectant pas toutes les contraintes de séparation. Pour cela, si l'on décide de normaliser les valeurs possibles de la fonction fitness entre 0 et 1, on pourra fixer un seuil (par exemple 0.5) séparant les éléments admissibles (de fitness supérieure à 0.5) des éléments non admissibles (de fitness inférieure à 0.5).

4.3.5 L'opérateur de sélection : $SELECTION[NPOP, POP]$

A l'inverse d'autres techniques d'optimisation, les algorithmes génétiques ne requièrent pas d'hypothèse particulière sur la régularité de la fonction objectif. L'algorithme génétique n'utilise notamment pas les dérivées successives de la fonction objectif, ce qui rend très vaste son domaine d'application. Aucune hypothèse

sur la continuité de la fonction objectif n'est requise. Néanmoins, dans la pratique, les algorithmes génétiques sont sensibles à la régularité de la fonction objectif.

Le peu d'hypothèses requises sur la fonction objectif permet de traiter des problèmes très complexes. La fonction objectif peut ainsi être le résultat d'une simulation.

La sélection permet d'identifier statistiquement les meilleurs individus d'une population et d'éliminer les mauvais. On trouve dans la littérature un nombre important de principes de sélection plus ou moins adaptés aux problèmes qu'ils traitent. Dans le cadre de notre travail, les deux principes de sélection suivants ont été envisagés :

- *Roulette wheel selection* [Gol89c] ;
- *Stochastic remainder without replacement selection* [Gol89c] ;

Le principe de *Roulette wheel selection* consiste à associer à chaque individu un segment dont la longueur est proportionnelle à sa fitness. On reproduit ici le principe de tirage aléatoire utilisé dans les roulettes de casinos avec une structure linéaire. Ces segments sont ensuite concaténés sur un axe que l'on normalise entre 0 et 1. On tire alors un nombre aléatoire de distribution uniforme entre 0 et 1, puis on "regarde" quel est le segment sélectionné. Avec ce principe, les grands segments, c'est-à-dire les bons individus, seront plus souvent adressés que les petits. Lorsque la dimension de la population est réduite, il est difficile d'obtenir en pratique l'espérance mathématique de sélection en raison du peu de tirages effectués. Un biais de sélection plus ou moins fort existe suivant la dimension de la population.

La *Stochastic remainder without replacement selection* évite ce problème et donne de bons résultats pour nos applications. Décrivons ce principe de sélection :

- Pour chaque élément i , on calcule le rapport r_i de sa fitness sur la moyenne des fitness.
- Soit $e(r_i)$ la partie entière de r_i , chaque élément est reproduit exactement $e(r_i)$ fois.
- La *roulette wheel selection* précédemment décrite est appliquée sur les individus affectés des fitness $r_i - e(r_i)$.

Compte-tenu du fait que des faibles populations seront utilisées par la suite, ce principe de sélection s'avèrera le plus efficace dans les applications pratiques et sera donc utilisé par la suite.

SELECTION[$NPOP$, POP]

- $moy = MOY(POP.fitness)$ (MOY effectue la moyenne des réels d'une liste)
- $som = SOMME(POP.fitness - ENT[POP.fitness/moy])$ (somme des fitness résiduelles, ENT définit la partie entière d'un réel).
- Pour i de 1 à K répéter :
 - Rajouter $ENT[POP(i).fitness/moy]$ fois $POP(i)$ dans la liste $NPOP$.
 - Tant que $NPOP$ n'est pas pleine :

- Choisir δ aléatoirement dans $[0, som]$ (distribution uniforme).
- $j = 0$
- Répéter tant que $(\delta \geq 0)$:
 - $j = j + 1$
 - $\delta = \delta - (POP(j).fitness - ENT[POP[j].fitness/moy])$
- Ajouter $POP(j)$ dans $NPOP$

La diversité de la population doit être entretenue au cours des générations afin de parcourir le plus largement possible l'espace d'états. C'est le rôle des opérateurs de croisement et de mutation.

4.3.6 Opérateur de Croisement : $CROISEMENT[NPOP]$

Le croisement a pour but d'enrichir la diversité de la population en manipulant la structure des chromosomes. Classiquement, les croisements sont envisagés avec deux parents et génèrent deux enfants.

$CROISEMENT[NPOP]$

- $n_c = ENT[P_c \times K/2]$
- Répéter n_c fois
 - Choisir aléatoirement 2 indices i et j dans $[1, K]$ (distribution uniforme) tels que $i \neq j$.
 - $(NPOP(i).data, NPOP(j).data) = CROISER[NPOP(i).data, NPOP(j).data]$

Problèmes discrets

Initialement, le croisement associé au codage par chaînes de bits est le croisement à découpage de chromosomes (slicing crossover). Pour effectuer ce type de croisement sur des chromosomes constitués de m gènes, on tire aléatoirement (distribution uniforme) une position dans chacun des parents. On échange ensuite les deux sous-chaînes terminales de chacun des deux chromosomes, ce qui produit deux enfants C_1 et C_2 (voir figure 4.2).

On peut étendre ce principe en découpant le chromosome non pas en 2 sous-chaînes mais en 3, 4, etc [BG91]. (voir figure 4.3).

Ce type de croisement à découpage de chromosomes est très efficace pour les problèmes discrets.

Pour nos applications, étant donné que l'ordre des variables n'a pas d'importance, on utilisera le croisement suivant :

$CROISER[POP(i).data, POP(j).data]$

- Pour l de 1 à m (où m est la taille de $POP(i).data$)
 - Choisir aléatoirement (distribution uniforme) $h = i$ ou $h = j$
 - $buf_i = POP(h).data(l)$
 - Si $(h = i)$, $buf_j = POP(j).data(l)$, sinon, $buf_j = POP(i).data(l)$

FIGURE 4.2 – Slicing crossover

FIGURE 4.3 – Slicing crossover à 2 points

FIGURE 4.4 – Croisement barycentrique

$$- POP(i).data(l) = buf_i, POP(j).data(l) = buf_j$$

Avec ce type de croisement, l'ordre des variables n'a plus d'importance. Le nombre de points de croisement peut valoir de 0 à $m - 1$.

Problèmes continus

Pour les problèmes continus, un croisement “barycentrique” est souvent utilisé : deux gènes $P_1(i)$ et $P_2(i)$ sont sélectionnés dans chacun des parents à la même position i . Ils définissent deux nouveaux gènes par pondération : deux nouveaux points sont créés sur la droite qui relie ces derniers. On crée ainsi $C_1(i)$ et $C_2(i)$ de la façon suivante :

$$\begin{cases} C_1(i) = \alpha P_1(i) + (1 - \alpha) P_2(i) \\ C_2(i) = (1 - \alpha) P_1(i) + \alpha P_2(i) \end{cases}$$

où α est un coefficient de pondération aléatoire adapté au domaine d'extension des gènes : il n'est pas nécessairement compris entre 0 et 1. Dans les applications, il prendra des valeurs aléatoires (distribution uniforme) dans l'intervalle $[-0.5, 1.5]$ ce qui permet de générer des points entre, ou à l'extérieur des deux gènes considérés.

Dans le cas particulier d'un chromosome matriciel constitué par la concaténation de vecteurs (ce sera le cas pour la résolution de conflits), on peut étendre ce principe de croisement aux vecteurs constituant les gènes (voir figure 4.4) :

$$\begin{cases} \vec{C}_1(i) = \alpha \vec{P}_1(i) + (1 - \alpha) \vec{P}_2(i) \\ \vec{C}_2(i) = (1 - \alpha) \vec{P}_1(i) + \alpha \vec{P}_2(i) \end{cases}$$

$CROISER[POP(i).data, POP(j).data]$

- Pour l de 1 à m (où m est la taille de $POP(i).data$)
- Choisir aléatoirement (distribution uniforme) α dans $[-0.5, 1.5]$

FIGURE 4.5 – Principe de l'opérateur de mutation

- $buf_i = \alpha POP(i).data(l) + (1 - \alpha) POP(j).data(l)$
- $buf_j = (1 - \alpha) POP(i).data(l) + \alpha POP(j).data(l)$
- $POP(i).data(l) = buf_i$
- $POP(j).data(l) = buf_j$

On peut imaginer et tester des opérateurs de croisement plus ou moins complexes sur un problème donné mais l'efficacité de ce dernier est souvent liée intrinsèquement au problème.

Pour le problème de résolution de conflits, le croisement barycentrique et le slicing crossover seront modifiés afin de permettre une convergence plus rapide vers des solutions admissibles.

4.3.7 Opérateur de mutation : $MUTATION[NPOP]$

L'opérateur de mutation apporte aux algorithmes génétiques la propriété d'ergodicité de parcours d'espace. Cette propriété indique que l'algorithme génétique sera susceptible d'atteindre tous les points de l'espace d'états, sans pour autant les parcourir tous dans le processus de résolution. Ainsi en toute rigueur, l'algorithme génétique peut converger sans croisement, et certaines implantations fonctionnent de cette manière [FOW66]. Les propriétés de convergence des algorithmes génétiques sont donc fortement dépendantes de cet opérateur sur le plan théorique.

Pour les problèmes discrets, l'opérateur de mutation consiste généralement à tirer aléatoirement (avec une distribution uniforme) un gène dans le chromosome et à le remplacer par une valeur aléatoire (voir figure 4.5). Si la notion de distance existe, cette valeur peut être choisie dans le voisinage de la valeur initiale.

Dans les problèmes continus, on procède un peu de la même manière en tirant aléatoirement (avec une distribution uniforme) un gène dans le chromosome, au-

FIGURE 4.6 – Exemple où les sélections classiques risquent de ne reproduire qu’un individu

quel on ajoute un bruit aléatoire, en veillant à ce que le gène résultant reste dans le domaine d’extension qui lui est propre.

MUTATION[*NPOP*]

- $n_m = ENT[P_m \times K]$
- Répéter n_m fois
 - Choisir i dans $[1, K]$ aléatoirement (distribution uniforme).
 - $NPOP(i).data = MODIFIER[NPOP(i).data]$

La procédure *MODIFIER* dépend de chaque application. Elle sera précisée par la suite dans chaque application.

4.4 Améliorations classiques

4.4.1 Introduction

Les processus de sélection présentés sont très sensibles aux écarts de fitness et dans certains cas, un très bon individu risque d’être reproduit trop souvent et peut même provoquer l’élimination complète de ses congénères ; on obtient alors une population homogène contenant un seul type d’individu. Ainsi, dans l’exemple de la figure 4.6 le second mode M_2 risque d’être le seul représentant pour la génération suivante et seule la mutation pourra aider à atteindre l’objectif global M_1 au prix de nombreux essais successifs.

Pour éviter ce comportement, il existe d’autres modes de sélection (*ranking*) ainsi que des principes (*scaling*, *sharing*) qui empêchent les individus “forts” d’éli-

miner complètement les plus “faibles”. Parmi les autres améliorations classiques, on envisagera la combinaison des algorithmes génétiques et du recuit simulé, ainsi qu’une méthode de sélection permettant d’obtenir les optima de Pareto d’un problème multi-critères.

4.4.2 Scaling : SCALING[NPOP]

Le *scaling* ou mise à l’échelle, modifie les fitness afin de réduire ou d’amplifier artificiellement les écarts entre les individus. Le processus de sélection n’opère plus sur la fitness réelle mais sur son image après scaling. Parmi les fonctions de scaling, on peut envisager le scaling linéaire et le scaling exponentiel. Soit f_r la fitness avant scaling, et f_s la fitness modifiée par le scaling.

Scaling linéaire

Dans ce cas, la fonction de scaling est définie de la façon suivante [Mic92] :

$$f_s = a f_r + b$$

$$a = \frac{\max' - \min'}{\max - \min}; b = \frac{\min' \cdot \max - \min \cdot \max'}{\max - \min}.$$

En règle générale, $a < 1$, ce qui permet de réduire les écarts de fitness et donc de favoriser l’exploration de l’espace. Ce scaling est statique par rapport au numéro de génération et pénalise la fin de convergence lorsque l’on désire favoriser les modes dominants.

Scaling exponentiel

Il est défini de la façon suivante [Mic92] (voir figure 4.7) :

$$f_s = (f_r)^{k(n)}$$

où n est la génération courante.

- Pour k proche de zéro, on réduit fortement les écarts de fitness ; aucun individu n’est vraiment favorisé et l’algorithme génétique se comporte comme un algorithme de recherche aléatoire et permet d’explorer l’espace.
- Pour k proche de 1 : le scaling est inopérant.
- Pour $k > 1$ les écarts sont exagérés et seuls les bons individus sont sélectionnés, ce qui produit l’émergence des modes.

Dans la pratique, on fait généralement varier k des faibles valeurs vers les fortes valeurs au cours des générations. Pour cela on peut utiliser la formule suivante :

$$k(n) = \left(\tan \left[\left(\frac{n}{N+1} \right) \frac{\pi}{2} \right] \right)^p$$

FIGURE 4.7 – Fonction de scaling exponentielle

n étant la génération courante, N le nombre total de générations prévues, p un paramètre à choisir. Le choix de $p = 0.1$ s'est avéré pertinent dans les applications (Nous ne détaillerons cependant pas ici les mesures de sensibilité faites sur quelques exemples pour évaluer ce choix). L'évolution de k en fonction de la génération n est donnée par la figure 4.8.

Ce dernier principe de scaling donne effectivement de meilleurs résultats sur nos problèmes que le scaling linéaire et sera donc systématiquement utilisé. Dans le cas des fonctions objectifs multi-modes présentant des optima quasi-équivalents, cette technique de scaling, en amplifiant les écarts de fitness en fin de convergence, va effectivement favoriser le mode dominant mais aussi masquer les modes sous-optimaux qui peuvent tout de même présenter un intérêt. Le scaling permet donc une bonne exploration de l'espace d'états mais ne favorise pas la répartition des individus sur les différents modes de la fonction objectif.

SCALING[*NPOP*, n]

- Pour i variant de 1 à K
- $NPOP(i).fitness = [NPOP(i).fitness]^{k(n)}$ (n est la génération)

4.4.3 Sharing

Introduction

L'objectif du sharing est de répartir sur chaque sommet de la fonction à optimiser, un nombre d'individus proportionnel à la fitness associée à ce sommet. La figure 4.9 présente deux exemples de répartition de populations dans le cas d'une fonction à cinq sommets : le premier sans sharing, le second avec sharing.

FIGURE 4.8 – Allure de l'évolution de k en fonction des générations

FIGURE 4.9 – Objectif du sharing

FIGURE 4.10 – Allure de $S(\frac{d}{\sigma_{share}})$ **Principe**

Le sharing consiste à modifier la fitness utilisée par le processus de sélection. Pour éviter le rassemblement des individus autour d'un sommet dominant, le sharing pénalise les fitness en fonction du taux d'agrégation de la population dans le voisinage d'un individu. Il requiert l'introduction d'une notion de distance. Dans la pratique, il faut définir une distance indiquant la dissimilarité entre deux individus. Cette distance est alors utilisée pour calculer la nouvelle fitness de la façon suivante :

$$f'_i = \frac{f_i}{m'_i}; m'_i = \sum_{j=1}^{j=N} S(d(x_i, x_j))$$

avec

$$S(d) = 1 - \left(\frac{d}{\sigma_{share}} \right)^\alpha \quad \text{si } d < \sigma_{share}$$

$$S(d) = 0 \quad \text{si } d > \sigma_{share}$$

Le paramètre σ_{share} permet de délimiter le voisinage d'un point et dépend du problème traité. La figure 4.10 donne l'allure de $S(d)$ pour différentes valeurs de α . Suivant la valeur donnée à α , le sharing sera plus ou moins efficace. Ainsi pour $\alpha < 1$, on pénalise les groupes très agglomérés.

Dans la pratique, ce type de sharing donne effectivement de bons résultats mais au prix de N^2 calculs de distances entre chromosomes à chaque génération pour une population de taille N . Or les algorithmes génétiques induisent une complexité en N sans sharing et le fait de passer en N^2 peut être pénalisant, notamment pour N grand.

Pour réduire la complexité, on utilise un sharing “clusterisé”.

4.4.4 Sharing clusterisé : *SHARING(NPOP)*

Ce type de sharing est décrit en détail dans [YG93]. On commence par identifier les différents “clusters” d’individus dans la population. Deux paramètres, d_{min} et d_{max} , doivent être définis respectivement pour fusionner des clusters ou en créer de nouveaux. Initialement, chaque individu de la population est considéré comme le centre d’un cluster. On applique alors successivement les deux principes suivants :

- si deux centres sont à une distance inférieure à d_{min} , on fusionne ces derniers dans un cluster unique dont le centre résultant est le barycentre des deux centres initiaux.
- un nouvel individu est agrégé à un cluster si sa distance au centre le plus proche est inférieure à d_{max} . Dans ce cas, on recalcule le centre du cluster global. Sinon, on crée un nouveau cluster contenant ce seul individu.

Ce principe de fusion-agrégation permet d’obtenir un nombre de clusters fluctuant avec la répartition des individus dans l’espace d’états. On applique ensuite le principe de sharing en modifiant les fitness de la façon suivante :

$$f'_i = \frac{f_i}{m'_i}; m'_i = l_c \left(1 - \left(\frac{d_{ic}}{2d_{max}} \right)^\alpha \right);$$

avec

- l_c : nombre d’individus contenus dans le cluster auquel appartient l’individu i .
- α : coefficient de sensibilité (pris égal à 2 dans les applications)
- d_{ic} : distance entre l’individu i et le centre du cluster c .

Ce type de sharing induit une complexité en $O(N \log N)$ [YG93] pour des résultats tout à fait comparables à ceux fournis par le sharing classique. Dans la pratique, on remarque que le réglage des coefficients d_{min} et d_{max} est assez délicat car l’efficacité de ces derniers dépend essentiellement de la connaissance *a priori* des distances inter-modes dans l’espace d’états, distance qu’il est très difficile d’estimer. Néanmoins, ce type de sharing sera utilisé pour la résolution de conflits en raison du gain de temps qu’il permet.

L’implémentation du sharing clusterisé requiert la création en début d’algorithme d’une liste de données *CLUS* constituées de trois champs :

- *CLUS.data* contenant le barycentre du cluster.
- *CLUS.l_c* contenant le nombre d’éléments du cluster.
- *CLUS.liste* contenant la liste des indices des éléments du cluster.

On crée également une liste d’entiers NUM_{clus} , de taille K qui contiendra pour chaque indice i le numéro du cluster dans lequel il se trouve. Le mécanisme de la procédure *CLUSTER(CLUS, NUM_{clus}, NPOP)* qui actualise ces listes à chaque génération a été décrit précédemment. On en trouvera un exemple détaillé dans [YG93].

SHARING(NPOP)

- $CLUSTER(CLUS, NUM_{clus}, NPOP)$
- Pour i de 1 à K
 - $j = NUM_{clus}(i)$
 - $buf = DIST[POP(i).data, CLUS[j].data]$ (La fonction $DIST$ retourne la distance entre deux données)
 - $buf = CLUS[j].l_c \times (1 - (buf/d_{max})^2)$
 - $NPOP(i).data = NPOP(i)/buf$

4.4.5 Elitisme

L'élitisme a pour but de préserver à chaque génération le meilleur élément de la population, et ce, afin d'éviter qu'un opérateur de croisement ou de mutation ne le détruise. Après chaque génération, on relève l'indice i_{meil} du meilleur élément de population. Dans les procédures *CROISEMENT* et *MUTATION*, lorsque cet indice i_{meil} est rencontré, un nouveau tirage aléatoire (de distribution uniforme sur la population) est effectué.

L'élitisme sera systématiquement utilisé dans les applications.

4.4.6 Algorithmes génétiques et recuit simulé

Introduction

Les algorithmes génétiques et le recuit simulé étant deux techniques d'optimisation stochastique travaillant sur les mêmes types de problèmes, il est logique de chercher à les associer afin de tirer partie de leurs avantages respectifs. Après plusieurs évaluations de ces deux techniques sur les mêmes problèmes test, on remarque que le recuit simulé converge généralement plus vite vers la solution optimale lorsque le problème est de taille raisonnable. Toutefois, il ne donne qu'une solution dans le cas des problèmes multi-modes. Ceci confirme les résultats donnés dans [IR92b]. A l'inverse, les algorithmes génétiques fournissent plusieurs solutions quasi-optimales mais au prix d'un temps de convergence généralement plus long. Il semble alors naturel d'associer ces deux techniques afin d'améliorer la convergence des algorithmes génétiques.

Il y a eu de nombreuses tentatives de fusion entre les algorithmes génétiques et le recuit simulé, les travaux les plus intéressants étant ceux de Mahfoud et de Goldberg [MG92].

Principe du croisement avec recuit simulé

Pour appliquer ce principe de croisement, on commence par sélectionner deux parents P_1 et P_2 dans la population (voir figure 4.11). On applique ensuite l'opérateur de croisement classique qui génère deux enfants C_1 et C_2 . Un tournoi est alors effectué entre les parents et les enfants pour lequel les deux vainqueurs sont

FIGURE 4.11 – Principe du croisement avec recuit simulé

sélectionnés par le schéma de recuit suivant. On considère l'individu C_1 . On tire ensuite aléatoirement un des deux parents, soit P_i ce parent :

- si C_1 est meilleur que $P_i \Rightarrow C_1$ est sélectionné.
- sinon C_1 est sélectionné avec la probabilité :

$$P = e^{-\left(\frac{|f_{C_1} - f_{P_i}|}{c_n}\right)}$$

où $c(n)$ est un coefficient décroissant en fonction de la génération courante (n).

On fait de même pour C_2 avec le parent restant et l'on détermine ainsi deux individus C'_1 et C'_2 .

L'évolution de la variable $C(n)$ se fait de la façon suivante. On utilise un schéma de recuit standard géométrique à un palier de basculement. Pratiquement, on calcule trois "températures" dont les valeurs dépendent de la connaissance des écarts *min* et *max* des fitness de la population initiale.

$$\left\{ \begin{array}{lll} C_s = -\frac{\Delta f_{max}}{\ln\left(\frac{1}{k-1}\right)} & k = 0.75 & \text{"Température initiale"} \\ C_x = -\frac{\Delta f_{max}}{\ln\left(\frac{1}{k-1}\right)} & k = 0.99 & \text{"Température de basculement"} \\ C_f = -\frac{\Delta f_{min}}{\ln\left(\frac{1}{k-1}\right)} & k = 0.99 & \text{"Température finale"} \end{array} \right.$$

où Δf_{min} , Δf_{max} représentent les écarts minimum et maximum des fitness de la population initiale. Le schéma géométrique fait évoluer la température courante de la façon suivante :

$$\begin{cases} C_0 = C_s \\ C_{n+1} = \alpha_1 C_n \text{ pour } C_s > C_n > C_x; \\ C_{n+1} = \alpha_2 C_n \text{ pour } C_x > C_n > C_f; \end{cases}$$

avec $0 < \alpha_1 < \alpha_2 < 1$.

Pour chaque palier, on calcule le nombre d'itérations de stabilisation à l'aide des formules :

$$N_1 = \frac{\ln\left(\frac{C_x}{C_s}\right)}{\ln \alpha_1} \quad N_2 = \frac{\ln\left(\frac{C_f}{C_x}\right)}{\ln \alpha_2}$$

Ces deux formules nous permettent de calculer le nombre total de générations pour un problème donné.

Ce même principe de recuit simulé a été essayé sur l'opérateur de mutation en faisant un schéma de recuit entre l'individu muté et l'individu d'origine mais il ne produit pas l'effet escompté. En effet, on peut supposer que dans ce cas, le recuit simulé réduit le brassage de la population provoqué par la mutation, en limitant l'espace exploré aux zones qui améliorent statistiquement la fitness, et en "interdisant" les domaines qui la dégradent. L'exploration du domaine admissible est fragilisée. Ce constat sur les problèmes que nous avons testés ne permet pas de généraliser.

En outre, pour le problème de résolution de conflits, la convergence se faisant en une vingtaine de générations seulement, l'introduction du recuit dans l'algorithme génétique ne s'avèrera pas pertinente et sera donc abandonnée. C'est pourquoi nous ne détaillerons pas la procédure.

4.4.7 Recherche des optima de Pareto

Dans la partie 1.4, nous avons abordé l'aspect subjectif de certains critères tels que le choix de l'avion à dévier dans le problème de résolution de conflits. En fait, dans ce problème multi-agents, chaque avion cherche à optimiser son critère (il souhaite être dévié le moins possible, en évitant les autres avions). Dans ce cadre, on peut s'intéresser à chercher les optima de Pareto du problème multi-critères ainsi défini.

Dominance au sens de Pareto

La notion de dominance au sens de Pareto se définit comme suit : on dit que le point A domine le point B si, $\forall i, f_i(A) \geq f_i(B)$ (avec au moins une inégalité stricte), où les f_i représentent les critères à maximiser. L'ensemble des points qui ne sont dominés par aucun autre point forme la surface de Pareto. Tout point de la surface de Pareto est "optimal", dans la mesure où on ne peut améliorer la valeur d'un critère pour ce point sans diminuer la valeur d'au moins un autre critère.

On peut construire un algorithme génétique permettant de trouver la surface de Pareto d'un problème multi-critères. Pour ce, il faut répartir la population de l'algorithme génétique sur la dite surface.

Modification de l'opérateur de sélection

La technique employée dérive directement des travaux de Jeffrey Horn et Nicholas Nafpliotis ([HN93]). Au lieu de travailler sur une fonction d'évaluation scalaire, on a désormais une fonction d'évaluation vectorielle (chaque critère est une variable du vecteur). En conséquence, le principal changement induit concerne le processus de sélection³.

On introduit une variante de la notion de dominance que l'on définit ainsi : l'élément E_i domine E_j si le nombre des valeurs contenues dans son vecteur d'adaptation qui sont supérieures aux valeurs correspondantes dans E_j dépasse un certain seuil⁴. La technique proposée pour effectuer la sélection est alors simple : on tire deux individus au hasard ainsi qu'une sous-population⁵ à laquelle ils n'appartiennent pas et qui va servir à les comparer.

Trois cas se présentent alors :

- si le premier élément domine tous ceux de la sous-population, et que ce n'est pas le cas pour le second, alors le premier sera sélectionné.
- inversement, si seul le second élément domine l'ensemble de la sous-population, alors c'est lui qui sera conservé.
- il reste deux possibilités : soit les deux sont dominés, soit les deux dominent.

On ne peut se prononcer sans ajouter un autre test, c'est pourquoi dans ce cas il est fait usage d'un nouveau type de sharing qui opère sur l'espace objectif.

Le sharing va conduire à sélectionner celui des deux individus qui a le moins de voisins proches, autrement dit, on élimine celui qui se situe dans une zone d'agrégation pour conserver celui qui est dans une région moins dense.

Le terme *voisin proche* n'a pas de signification précise mais il est possible de définir un voisinage (aussi appelé niche) correct en se servant de la distance de Holder :

$$d_H(E_i, E_j) = \left(\sum_{k=1}^n |f_i^k - f_j^k|^p \right)^{\frac{1}{p}}$$

f_i^k désignant la k -ième composante du vecteur adaptation de l'élément i . Le paramètre p permet de faire varier la forme et la taille du voisinage (dans l'application présentée en 4.6.10, $p = 2$). A l'intérieur des voisinages ainsi définis dans l'espace objectif, il suffit de compter le nombre d'individus pour favoriser les zones les moins denses, et de cette façon, maintenir la diversité de la population.

3. En ce qui concerne les autres opérateurs de base, à savoir le croisement et la mutation, il n'y a aucune modification, car ceux-ci travaillent dans l'espace d'état et non dans l'espace résultat.

4. 80% dans l'application de la partie 4.6.10

5. La sous-population tirée aura une taille proportionnelle à celle de la population de départ (généralement autour de 10%). Seule une partie des individus est ainsi utilisée, ce qui permet de réduire le temps de calcul.

4.5. FONCTIONS PARTIELLEMENT SÉPARABLES : CROISEMENT ADAPTÉ¹²³

La recherche des optima de Pareto pour le problème de résolution de conflits sera tentée mais abandonnée car trop coûteuse en temps de calcul pour un système en temps réel (une évaluation de ce temps est donnée dans la partie 4.6.10). On privilégiera une recherche avec sharing clusterisé qui permet de dégager plusieurs solutions proches de la meilleure solution.

4.4.8 L'algorithme génétique amélioré retenu

- *INITIALISER*[POP]
- Répéter pour k variant de 0 à N :
 1. *EVALUER*[POP]
 2. *SCALING*[POP, n]
 3. *SHARING*[POP]
 4. *SELECTION*[NPOP, POP]
 5. *CROISEMENT*[NPOP]
 6. *MUTATION*[NPOP]
 7. $POP = NPOP$
- *EVALUER*[POP]
- *MEILLEUR*[POP]

Cet algorithme sera utilisé dans les applications numériques. Pour chaque algorithme, nous préciserons :

- Le nombre de générations
- Le nombre d'éléments de population
- Pour le sharing : d_{min} , d_{max} ainsi que la distance utilisée.
- Le type de croisement utilisé (slicing ou barycentrique)
- Le générateur de bruit pour la mutation

Les pourcentages de croisement P_c et de mutation P_m seront fixés pour toutes les applications suivantes à 0.6 et 0.15. Des tests effectués avec des probabilités de croisement variant entre 0.4 et 0.8 et des probabilités de mutation variant entre 0.10 et 0.20 semblent montrer que l'AG est peu sensible aux variations de ces paramètres pour des valeurs ainsi fixées (il convient néanmoins de rappeler que les tests n'ont été effectués que sur les fonctions utilisées dans les applications présentées).

4.5 Fonctions partiellement séparables : croisement adapté

4.5.1 Introduction

Dans beaucoup de problèmes d'optimisation, la fonction à optimiser dépend de nombreuses variables, et peut se décomposer comme somme de sous-fonctions prenant en compte une partie des variables seulement. Les algorithmes génétiques s'avèrent être de bons outils d'optimisation globale en raison de leur capacité à

sortir des minima locaux. Néanmoins, leurs performances sont souvent pénalisées par le caractère très aléatoire des opérateurs de croisement et de mutation. Pour remédier à ce phénomène, certains évolutionnistes [SSR94, RSS95] utilisent des heuristiques qui permettent de favoriser les “bons” croisements ou les “bonnes” mutations et d’éviter les “mauvaises”. Dans ce paragraphe, on introduit un opérateur de croisement utilisable pour des problèmes où la fonction à optimiser peut se décomposer en somme de sous-fonctions positives ne dépendant pas de toutes les variables du problème.

Après une présentation formelle du type de problèmes auxquels s’adresse cette méthode, nous présenterons l’opérateur de croisement. Une illustration de l’efficacité de cette méthode sera ensuite proposée au travers de plusieurs exemples simples comportant jusqu’à une cinquantaine de variables. On trouvera également dans [DAN94] une application de cet outil au problème du Voyageur de Commerce.

4.5.2 Problèmes concernés

Il s’agit de problèmes ou fonctions partiellement séparables [GT82], à savoir des problèmes où la fonction F à optimiser dépend de plusieurs variables x_1, x_2, \dots, x_n , et peut se décomposer en somme de fonctions F_i positives ne dépendant pas de toutes les variables. On remarquera que bon nombre de fonctions multiplicatives peuvent se transformer en fonction additives grâce à des transformations simples. On s’intéresse donc aux fonctions pouvant s’écrire :

$$F(x_1, x_2, \dots, x_n) = \sum_{i=1}^m F_i(x_{j_1}, x_{j_2}, \dots, x_{j_{n_i}})$$

Soit E_k l’ensemble des indices i tels que x_k soit une variable de F_i :

$$E_k = \{i/x_k \in \text{variables de } F_i\}$$

Pour définir notre opérateur de croisement, nous allons introduire pour chaque variable x_k sa “fitness locale” $G_k(x_1, x_2, \dots, x_n)$ définie comme suit :

$$G_k(x_1, x_2, \dots, x_n) = \sum_{i \in E_k} F_i(x_{j_1}, x_{j_2}, \dots, x_{j_{n_i}})$$

La fitness locale associée à une variable isole sa contribution. Le but de ce paragraphe est de proposer un opérateur de croisement utilisant la fitness locale qui permet de réduire la taille des populations et de converger plus rapidement vers la solution optimale. Lorsque la fonction à optimiser comporte plusieurs optima, l’utilisation simultanée du sharing et de cet opérateur s’avère souvent efficace.

4.5.3 L’opérateur de croisement adapté

Le codage que nous adopterons pour appliquer notre croisement consiste à représenter le chromosome par la liste brute des variables du problème. S’il s’agit

4.5. FONCTIONS PARTIELLEMENT SÉPARABLES : CROISEMENT ADAPTÉ 125

de bits, nous aurons une liste de bits, s'il s'agit de variables réelles, nous aurons une liste de réels. On pourra éventuellement avoir un panachage de différents types de variables. L'idée intuitive est la suivante : dans le cas d'un problème complètement séparable, optimiser le problème global peut se réduire à l'optimisation de chaque variable séparément. Nous essayons d'adapter ce raisonnement au cas de fonctions partiellement séparables. La technique que nous proposons consiste donc à retenir pour chaque variable, lors de la conception des fils, celle des deux parents qui a la meilleur fitness locale, ceci à un facteur δ près. Par exemple, dans le cas où l'on cherche un minimum pour F , pour la $k^{\text{ième}}$ variable, si :

$$G_k(\text{pere}_1) < G_k(\text{pere}_2) - \delta$$

alors le fils 1 se verra affecter la variable x_k du père 1. Si par contre :

$$G_k(\text{pere}_1) > G_k(\text{pere}_2) + \delta$$

il se verra affecter la variable x_k du père 2. Si enfin :

$$\|G_k(\text{pere}_1) - G_k(\text{pere}_2)\| \leq \delta$$

la variable x_k du fils 1 sera choisie aléatoirement (avec une distribution uniforme). Dans ce dernier cas on peut également choisir la variable du fils en utilisant un croisement barycentrique (lorsque cela sera le cas, nous le préciserons).

Lors du croisement, on souhaite créer deux fils à partir de deux parents. Lorsque δ est faible, si l'on applique la même stratégie pour le fils 2 que celle décrite ci-dessus pour le fils 1, les deux fils risquent de se ressembler fortement. Ceci n'est pas nécessairement souhaitable. On peut donc pour le deuxième fils choisir une autre valeur de δ , ce qui permet d'être plus ou moins déterministe, ou alors, comme pour les techniques de croisement classiques, choisir le complémentaire du fils 1, à savoir affecter au fils 2 la variable des deux pères non affectée au fils 1.

Dans les applications, nous préciserons les choix retenus.

On peut remarquer qu'il est facile d'introduire un opérateur de mutation qui s'appuie sur le même principe et modifie avec une plus forte probabilité les variables ayant une mauvaise fitness locale.

4.5.4 Application à un polynôme

Pour illustrer l'intérêt de notre opérateur de croisement, considérons le polynôme suivant :

$$F(x_1, x_2, \dots, x_n) = \sum_{i \neq j} (x_i - x_j)^2$$

On définit les fitness locales comme suit :

$$G_k(x_1, x_2, \dots, x_n) = \sum_{i \neq k} (x_i - x_k)^2$$

Croisement	classique	adapté
Nb min de gens	94	20
Nb moy de gens	131.6	38.27
Nb max de gens	180	55
Temps min (en sec)	55	12
Temps moy (en sec)	78.08	21.76
Temps max (en sec)	110	30

TABLE 4.1 – Résultats numériques sans sharing, $n = 50$.

On cherche le minimum de notre polynôme sur l'ensemble des entiers (x_1, x_2, \dots, x_n) de $[0, 20]^n$.

Dans un premier temps, pour tester la valeur de ce nouvel opérateur de croisement, nous avons effectué deux séries de tests. Pour cela, nous avons choisi $n = 50$. Les deux séries de tests n'utilisent pas de sharing et ne se différencient que par l'utilisation ou non de notre croisement. La première série de simulations utilise donc une méthode de croisement classique (Le Slicing Crossover décrit précédemment). La deuxième utilise notre nouvel opérateur. La valeur de δ sera fixée à n , la taille du problème. Les deux enfants seront construits de la même façon. Le choix de δ est empirique. Pour δ trop petit, l'algorithme converge très rapidement vers un optimum local. Pour δ trop grand, l'opérateur adapté devient inefficace.

L'opérateur de mutation retenu dans ce cas consiste à choisir au hasard l'une des variables de l'individu, et à lui ajouter ou retrancher 1 (au hasard également). Lorsque la variable ainsi modifiée sort de l'intervalle $[0, 20]$, elle est tronquée.

Pour chacune des deux séries de tests (chaque test est répété 100 fois avec un générateur de nombre aléatoire différent à chaque fois), nous mesurons le temps minimum, moyen et maximum mis par l'algorithme pour trouver une solution optimale, le nombre de générations minimum, moyen et maximum correspondants. Les simulations ont été faites sur une HP 720.

La taille de la population est fixée à 100. Le recuit simulé décrit précédemment est utilisé dans les deux cas. Pour des raisons de normalisation, au lieu de minimiser F , nous avons maximisé $\frac{1}{1+F}$.

Les résultats sont répertoriés dans le tableau 4.1.

On montre ainsi que l'opérateur de croisement introduit permet de diviser le temps de calcul par 3.5, ce qui confirme l'intérêt de cet opérateur.

Afin de montrer comment cette technique de croisement s'adapte avec le sharing, nous avons effectué un autre test comparatif. Cette fois, l'entier n vaut 20.

Les choix faits pour le sharing sont les suivants : Si $X = (x_1, \dots, x_n)$ et $Y = (y_1, \dots, y_n)$ sont deux éléments de population :

$$DISTANCE[X, Y] = |x_1 - y_1| + \dots + |x_n - y_n|$$

On choisit empiriquement $d_{max} = n$ et $d_{min} = n/2$.

4.5. FONCTIONS PARTIELLEMENT SÉPARABLES : CROISEMENT ADAPTÉ 127

FIGURE 4.12 – Meilleure fitness en fonction de la génération courante, nouveau croisement avec sharing (courbe du haut), croisement classique avec sharing (courbe du bas).

Il y a trivialement 21 optima globaux à ce problème. La taille de l'espace de recherche est 21^{20} , soit plus de 10^{27} . Nous avons effectué deux simulations. La première fait appel à une méthode de croisement classique avec sharing et la deuxième utilise la méthode de croisement décrite ci-dessus avec sharing.

Les paramètres des simulations sont les suivants :

Taille de la population : 600
Nombre de générations : 100

Le temps de calcul moyen est de l'ordre de 300 secondes.

La figure 4.12 nous donne l'évolution du meilleur élément de population dans les deux simulations. On observe que la convergence vers un optimum est beaucoup plus rapide lorsque l'on utilise le croisement adapté. L'optimum est alors atteint dès la 22^{ème} génération au lieu de la 63^{ème} lorsque l'on utilise le croisement classique.

Le résultat le plus intéressant apparaît dans la figure 4.13 donnant le nombre de solutions optimales obtenues dans les deux simulations. On observe alors que le croisement adapté a joué un rôle important dans la deuxième simulation puisque les 21 optima globaux sont obtenus, alors qu'avec le croisement classique, seuls 4 optima sont obtenus.

FIGURE 4.13 – Nombre de solutions optimales en fonction de la génération courante, nouveau croisement avec sharing (courbe du haut), croisement classique avec sharing (courbe du bas).

4.5.5 Une fonction trompeuse

Dans la fonction précédente, si toutes les variables sauf une (x_i par exemple) sont fixées à l'optimum, la fitness locale G_i est optimale pour la valeur optimale de la fitness globale. Il serait intéressant de biaiser ce phénomène. Dans Z/nZ , on définit la distance $d_n(x, y)$ entre deux éléments x et y par le minimum entre $|x - y|$ et $n - |x - y|$. On peut alors définir pour (x_1, x_2, \dots, x_n) entiers de $[1, n]^n$:

$$F(x_1, x_2, \dots, x_n) = \alpha \sum_{i \neq j} d_n(x_i, x_j) + n \beta \sum_{i=1}^n d_n(x_i, i)$$

Si le couple (α, β) vaut $(1, 1 + \epsilon)$ (avec $\epsilon > 0$), la valeur minimale de la fitness est obtenue pour $x_i = x_j, \forall (i, j) \in [1, n]^2$. Si (α, β) vaut $(1 + \epsilon, 1)$, la valeur minimale de la fitness est obtenue pour $x_i = i, \forall i \in [1, n]$. On peut ainsi, en faisant varier légèrement le couple (α, β) changer radicalement le minimum de la fonction.

Les opérateurs de croisement classique et de mutation utilisés sont les mêmes que précédemment, ainsi que δ, d_{max} et d_{min} . Le recuit simulé n'est plus utilisé. Les autres paramètres sont :

n : 10

Taille de la population : 600

4.5. FONCTIONS PARTIELLEMENT SÉPARABLES : CROISEMENT ADAPTÉ 129

sharing	non	non	oui	oui
(α, β)	(1.001, 1)	(1, 1.001)	(1.001, 1)	(1, 1.001)
optimum trouvé	17/100	21/100	16/100	99/100
génération moyenne	26	22	53	51
génération min	14	14	25	18
génération max	35	32	111	201
optimum non trouvé	83/100	79/100	84/100	1/100

TABLE 4.2 – Résultats numériques avec le croisement classique.

sharing	non	non	oui	oui
(α, β)	(1.001, 1)	(1, 1.001)	(1.001, 1)	(1, 1.001)
optimum trouvé	100/100	0/100	100/100	6/100
génération moyenne	5		5	505
génération min	4		4	472
génération max	6		6	564
optimum non trouvé	0/100	100/100	0/100	94/100

TABLE 4.3 – Résultats numériques avec le croisement adapté.

Nombre de générations : 200

4 séries de 100 tests utilisant le croisement classique ont été effectuées :

- sans sharing avec $(\alpha, \beta) = (1, 1.001)$.
- sans sharing avec $(\alpha, \beta) = (1.001, 1)$.
- avec sharing avec $(\alpha, \beta) = (1, 1.001)$.
- avec sharing avec $(\alpha, \beta) = (1.001, 1)$.

Les résultats sont présentés dans le tableau 4.2 et montrent que l'utilisation du sharing permet d'obtenir presque toujours l'optimum $(\alpha, \beta) = (1, 1.001)$, c'est à dire une solution du type $x_i = x_j, \forall (i, j) \in [1, n]^2$. Il est à noter qu'il existe n solutions optimales dans ce cas, alors qu'il n'en existe qu'une dans le cas où $(\alpha, \beta) = (1.001, 1)$.

On peut définir les fitness locales comme suit :

$$G_k(x_1, x_2, \dots, x_n) = \alpha \sum_{i \neq k} d_n(x_i - x_k) + n \beta d_n(x_k, k)$$

Les résultats des 4 séries de 100 tests, pour le croisement adapté utilisant les fitness locales, sont reportés dans le tableau 4.3. On observe que le croisement adapté permet de converger très rapidement vers l'optimum pour $(\alpha, \beta) = (1.001, 1)$, c'est à dire pour la solution $x_i = i, \forall i \in [1, n]$, mais ceci se fait au détriment des performances de l'algorithme dans la recherche de l'optimum pour $(\alpha, \beta) = (1, 1.001)$.

sharing	non	non	oui	oui
(α, β)	(1.001, 1)	(1, 1.001)	(1.001, 1)	(1, 1.001)
optimum trouvé	64/100	34/100	100/100	100/100
génération moyenne	9	11	9	36
génération min	5	5	4	4
génération max	16	20	20	336
optimum non trouvé	36/100	66/100	0/100	0/100

TABLE 4.4 – Résultats numériques avec le croisement adapté modifié.

On peut alors imaginer de modifier les fitness locales de façon à donner moins d'importance au deuxième terme. Ainsi, si l'on prend :

$$G_k(x_1, x_2, \dots, x_n) = \alpha \sum_{i \neq k} d_n(x_i - x_k) + \frac{n}{A} \beta d_n(x_k, k)$$

avec $A = 5$ les résultats (tableau 4.4) montrent que l'algorithme a 100% de réussite dans tous les cas.

Cette application est intéressante car elle montre que l'utilisation de l'opérateur de croisement adapté peut dégrader la convergence de l'algorithme dans certains cas. Cette application permet également de montrer que l'opérateur de croisement adapté peut être modulé afin de donner plus ou moins d'importance à certains termes.

4.5.6 Application à la fonction de Corana

Cette fonction est détaillée dans [CMMR87]. Les résultats obtenus par Ingber [IR92b] avec VFSR⁶ (ou ASA⁷) sont utilisés pour comparer les performances des AGs et du recuit simulé. La fonction de Corana prend la forme suivante :

$$F(x_1, \dots, x_N) = \sum_{i=1}^N \begin{cases} 0.15 d_i (0.05 S(z_i) + z_i)^2 & \text{si } |x_i - z_i| < 0.05 \\ d_i x_i^2 & \text{sinon} \end{cases}$$

$$z_i = 0.2 [|x_i/0.2| + 0.49999] S(x_i)$$

$$S(z_i) = \begin{cases} 1 & \text{if } z_i > 0 \\ 0 & \text{if } z_i = 0 \\ -1 & \text{if } z_i < 0 \end{cases}$$

$$d_i \text{ modulo } 4 = \{1.0, 1000.0, 10.0, 100.0\}$$

Sur le compact $[-10000, 10000]^N$, cette fonction a 10^{5N} optima locaux. D'après L. Ingber, cette fonction est un des meilleurs tests pour évaluer l'efficacité d'un algorithme d'optimisation globale. Il apparait donc intéressant de comparer les résul-

6. Very Fast Simulated Reannealing

7. Adaptive Simulated Annealing

tats obtenus avec VFSR à ceux obtenus avec un AG classique et avec le croisement adapté.

On notera que la fonction de Corana est complètement séparée.

Les paramètres de l'AG utilisés sont les suivants : l'opérateur de mutation consiste à rajouter un bruit gaussien de moyenne nulle et d'écart-type 1 à chacune des variables d'un élément de population. Le croisement classique est le croisement barycentrique. Pour le croisement adapté, δ est fixé à 0.

VFSR et les AGs classiques donnent de bons résultats sur cette fonction pour $N = 4$ et $N = 8$. Pour $N = 12$, le problème est plus difficile. Les paramètres par défaut de VFSR ne permettent plus à l'algorithme de converger.

Avec l'aide de L. Ingber, il a été possible de trouver des paramètres faisant converger VFSR pour $N = 12$ (grâce à une technique de "quenching" utilisée pour accélérer le schéma de recuit). Sur 10 tests exécutés, 5 ont donné le minimum global. L'AG classique a toujours donné le minimum global avant la génération 300 avec une population de 400 éléments. Cependant, lorsqu'il converge, VFSR est plus rapide que l'AG.

En augmentant la valeur de N pour comparer le comportement des deux algorithmes, on constate que pour $N = 20$, VFSR ne trouve plus le minimum global (sur 50 tests). Avec l'AG classique, on a pu trouver le minimum global jusqu'à $N = 24$, mais pas au-delà.

Pour appliquer l'AG avec les opérateurs adaptés, on définit trivialement la fitness locale comme suit :

$$G_i(x_i) = \begin{cases} 0.15 d_i (0.05 S(z_i) + z_i)^2 & \text{si } |x_i - z_i| < 0.05 \\ d_i x_i^2 & \text{sinon} \end{cases}$$

Puisque la fonction est totalement séparée, on pouvait s'attendre à ce que l'opérateur adapté se révèle particulièrement efficace, ce que confirment les tests : avec une population de 400 éléments sur 100 tests, l'AG utilisant le croisement adapté trouve toujours le minimum global avant la génération 100 pour $N = 1000$.

4.5.7 Application à la fonction de Griewank

Cette fonction ressemble à la précédente, mais a l'avantage pour nous de ne pas être séparable.

Elle est définie comme suit :

$$\forall (x_1, \dots, x_{10}) \in [-10000, 10000]^{10}, F(x_1, \dots, x_{10}) = \frac{1}{4000} \sum_{i=1}^{10} x_i^2 - \prod_{i=1}^{10} \cos\left(\frac{x_i}{\sqrt{i}}\right)$$

Cette fonction n'est pas représentable facilement mais on peut par exemple tracer $F(x + y, x + y, \dots, x + y)$ pour x variant entre $-\Pi$ et Π et y variant entre -20 et 20 (voir figure 4.14) et observer la projection sur un plan orthogonal à l'axe des x du résultat obtenu. On observe alors que la fonction a de nombreux minima locaux proches de 0, son minimum global.

Les paramètres de l'AG sont les suivants :

FIGURE 4.14 – Projection sur un plan orthogonal à l'axe des x de $F(x + y, x + y, \dots, x + y)$ pour $x \in [-\Pi, \Pi]$ et $y \in [-20, 20]$

4.5. FONCTIONS PARTIELLEMENT SÉPARABLES : CROISEMENT ADAPTÉ 133

type de croisement	min	max	moy	σ	dist à 0
classique	76	294	179.44	178.47	0.00
adapté	42	204	92.33	91.81	2.03
classique avec sharing	79	357	248.87	247.32	0.06
adapté avec sharing	42	186	96.57	95.95	0.01

TABLE 4.5 – 600 générations et 100 éléments de population

- Nombre de générations : 600
 - Taille de la population : 100
 - $DISTANCE[X, Y] = |x_1 - y_1| + \dots + |x_{10} - y_{10}|$
 - $d_{max} = 10, d_{min} = 5$
 - Croisement classique : barycentrique
 - Mutation : ajout d'un bruit gaussien à chaque variable de moyenne nulle et d'écart-type 1
 - Croisement adapté : $\delta = 10$ (On utilisera le croisement barycentrique lorsque la différence des deux fitness locales est inférieure à 10)
- On définit la fitness locale comme suit :

$$G_i(x_1, \dots, x_{10}) = \frac{1}{4000} x_i^2 - \prod_{i=1}^{10} \cos\left(\frac{x_i}{\sqrt{i}}\right)$$

4 séries de tests sont effectuées sur cette fonction :

- sans sharing avec un croisement classique de type barycentrique.
- sans sharing avec le croisement adapté.
- avec sharing avec un croisement classique de type barycentrique.
- avec sharing et le croisement adapté.

Dans chaque cas, l'algorithme génétique a été testé une centaine de fois avec 600 générations et 100 individus. On dira que l'algorithme génétique a convergé lorsqu'il a trouvé une solution, qui, par une méthode locale simple⁸ permet d'atteindre l'optimum global. En conséquence, on considèrera que l'algorithme a convergé dès que :

$$\forall i \in [0, 10], -\sqrt{i} \frac{\Pi}{2} \leq x_i \leq \sqrt{i} \frac{\Pi}{2}$$

Le tableau 4.5 donne le nombre minimal, maximal et moyen de générations nécessaires pour que l'algorithme génétique converge. Le tableau donne également l'écart-type de ce nombre de générations ainsi que la distance moyenne (en norme infinie) de la solution optimale trouvée à la solution optimale absolue, (0).

Les résultats ci-dessus montrent que la convergence de l'algorithme est 2 fois plus rapide avec le croisement adapté qu'avec un croisement classique. On observe

8. La méthode utilisée ici consiste à modifier successivement chaque variable suivant la valeur de la dérivée partielle de la fonction.

également que le sharing ralentit légèrement la convergence de l'algorithme quel que soit le croisement utilisé. Ce problème, difficile pour des méthodes d'optimisation locales et pour des algorithmes de recuit simulé (des tests ont été effectués avec *VFSR* [IR92a, Ing89] et n'ont pas convergé vers l'optimum⁹) semble facile pour l'algorithme génétique. Ainsi l'intérêt de l'opérateur de sharing n'est pas évident dans ce cas. Cependant, les performances de l'algorithme ne sont pas dégradées par l'opérateur de sharing lorsque le croisement adapté est utilisé. On observe même dans ce cas que l'opérateur de sharing permet d'obtenir une solution après convergence proche de 0 alors que sans l'opérateur de sharing, la meilleure solution après convergence reste éloignée de 0.

La figure 4.15 donne les valeurs moyennes des meilleures fitness suivant les 4 tests. Cette figure confirme le bon comportement de l'opérateur adapté avec sharing.

Dans chacune de ces 4 séries de tests, il est possible d'observer l'effet du croisement de la façon suivante : à chaque génération, on effectue 60 croisements, parmi ceux-ci, certains donnent soit un individu optimal (au sens défini précédemment), soit un individu acceptable, soit un individu qui sera immédiatement rejeté. Les courbes 4.16, 4.17, 4.18 et 4.19 donnent les répartitions de ces catégories pour un test extrait de chacune des 4 séries de tests.

On observe sur les 4 figures que lorsque l'algorithme génère des solutions optimales, il génère également des solutions rejetées (tellement mauvaises que l'opérateur de sélection ne les conserve pas). C'est la difficulté de ce problème (les solutions optimales sont proches de solutions très mauvaises). On observe sur la figure 4.17 que l'opérateur de croisement adapté génère un nombre très important de solutions rejetées. Il est intéressant de constater que l'opérateur de sharing permet d'atténuer cet effet.

4.6 Application au problème de résolution de conflits

4.6.1 Codage des données

Pour le problème de résolution de conflits, les variables t_0 , t_1 , t_2 et α de chaque avion sont disposées dans une matrice (voir figure 4.20) qui constitue un individu de la population. On appellera gènes les variables t_0 , t_1 , t_2 et α de chaque avion. Un élément de population contient donc $4n$ gènes.

4.6.2 Gestion des contraintes annexes : $VERIF[POP(i).data]$

Dans le modèle que nous avons défini partie 3, l'algorithme de résolution de conflits est réinitialisé pour chaque cluster d'avions toutes les Δ minutes. Il se peut donc que l'algorithme ait à prendre en compte des avions déjà engagés dans certaines manœuvres de résolution. Ces avions sont alors soumis à certaines

9. Il semble cependant que l'ajustement des paramètres de *VFSR* soit difficile, on peut craindre que l'absence de convergence résulte d'un manque d'expérience sur cet algorithme.

FIGURE 4.15 – Valeur moyenne de l'opposé de la meilleure fitness en fonction de la génération courante

FIGURE 4.16 – Nombre de croisements donnant une solution optimale, acceptable ou rejetée en fonction de la génération courante, croisement classique sans sharing

FIGURE 4.18 – Nombre de croisements donnant une solution optimale, acceptable ou rejetée en fonction de la génération courante, croisement classique avec sharing

FIGURE 4.20 – Codage d'un individu à n avions.

contraintes gérées *a priori* par une procédure chargée de vérifier leur respect après l'initialisation de la population et après chaque opérateur de croisement et de mutation. Lorsqu'un conflit à n avions est fourni à l'algorithme génétique, il se peut que certains gènes correspondant à des manœuvres déjà entamées ou qui doivent être entamées avant Δ , soient imposés à l'algorithme génétique. Ainsi, tous les gènes t_0, t_1, t_2 , fournis par l'optimisation précédente, dont la valeur est inférieure à Δ sont imposés dans l'algorithme et ne pourront plus être modifiés par la suite. De même, si $t_0 < \Delta$, alors l'angle α correspondant est fixé et ne pourra plus être modifié. Ces variables ne pourront donc pas prendre des valeurs quelconques.

Le temps est discrétisé (par pas de 12 secondes dans les applications) et les gènes t_0, t_1 et t_2 sont comptés en pas de temps. Un certain nombre de variables globales sont définies :

- dev_{min} est le temps de déviation minimal autorisé (il est peu raisonnable de dévier un avion pour 12 secondes seulement). On fixe dev_{min} à 5 pas de temps.
- dev_{max} est le temps de déviation maximal autorisé. Ce temps dépendra du nombre d'avions dans le conflit. En effet, un conflit à 10 avions peut nécessiter des déviations plus importantes qu'un conflit à 2 avions. On a choisi de fixer $dev_{max} = 2 \times n \times \frac{d}{V}$, où d est la séparation standard, V la vitesse de l'avion le plus lent et 2 l'inverse de $\sin(30)$. Ceci doit permettre à un avion de s'éloigner de n séparations standard de sa trajectoire.
- off_{min} est le temps d'offset minimal fixé à 10 pas de temps. Si le temps d'offset est trop court, l'offset ne se justifie plus.
- off_{max} est le temps d'offset maximal fixé à $H' = 1.3 \times H$ où H est l'horizon temporel (fixé à 20 minutes dans les applications). off_{max} est supérieur à H de façon à ne pas contraindre l'avion à terminer sa manœuvre avant la fin de l'horizon temporel. Lorsque l'avion n'est pas dévié, t_0, t_1 et t_2 sont fixés à H' .

On définira une procédure *VERIF* chargée de vérifier et corriger un élément de population afin qu'il respecte toutes les contraintes précédentes.

VERIF[*POP*(i).*data*]

Pour chaque avion d'un champs *POP*(i).*data* :

- Si t_0 n'est pas imposé :
 - Si $t_0 < \Delta$, alors $t_0 = \Delta$
 - Si $t_0 > H'$, alors $t_0 = H'$
 - Si $t_1 > H'$, alors $t_1 = H'$
 - Si $t_2 > H'$, alors $t_2 = H'$
 - Si $t_1 < t_0 + dev_{min}$, alors $\alpha = 0$ et $t_0 = t_1 = t_2 = H'$ (l'avion n'est pas dévié)
 - Si $t_1 > t_0 + dev_{max}$, alors $t_1 = t_0 + dev_{max}$
 - Si $t_2 < t_1 + off_{min}$, alors $t_2 = t_1$ (l'offset est supprimé)
- Si t_0 et α sont imposés mais t_1 n'est pas imposé :

- Si $t_1 < \Delta$, alors $t_1 = \Delta$
- Si $t_1 > H'$, alors $t_1 = H'$
- Si $t_2 > H'$, alors $t_2 = H'$
- Si $t_1 < t_0 + dev_{min}$, alors $t_1 = t_0 + dev_{min}$
- Si $t_1 > t_0 + dev_{max}$, alors $t_1 = t_0 + dev_{max}$
- Si $t_2 < t_1 + off_{min}$, alors $t_2 = t_1$ (l'offset est supprimé)
- Si t_0, t_1 et α sont imposés mais t_2 n'est pas imposé :
 - Si $t_2 < \Delta$, alors $t_2 = \Delta$
 - Si $t_2 > H'$, alors $t_2 = H'$
 - Si $t_2 < t_1 + off_{min}$, alors $t_2 = t_1 + off_{min}$
- Dans le dernier cas, t_0, t_1, t_2 et α sont imposés.

Pour la résolution de conflits, la procédure *VERIF* sera systématiquement appliquée à chaque élément de population dès lors qu'il est modifié. Ainsi, elle intervient dans les procédures *INITIALISER*, *CROISEMENT* et *MUTATION*.

INITIALISER(POP) est décrite partie 4.6.3.

Il convient de rajouter à la fin de la procédure *CROISEMENT[NPOP]* :

- $NPOP(i).data = VERIF[NPOP(i).data]$
- $NPOP(j).data = VERIF[NPOP(j).data]$

Il convient de rajouter à la fin de la procédure *MUTATION[NPOP]* :

- $NPOP(i).data = VERIF[MODIFIER[NPOP(i).data]]$

4.6.3 Initialisation de la population

On tire avec une distribution uniforme un réel ϵ dans $[0, 1]$.

- si $\epsilon \leq \frac{1}{3}$, $t_0 = H'$, $t_1 = H'$, $t_2 = H'$, et $\alpha = 0$ (l'avion n'est pas dévié).
- sinon, $t_0, t_1 - t_0$ et $t_2 - t_1$ sont choisis aléatoirement (distribution uniforme) dans $[0, H'/3]$, α est choisi aléatoirement (distribution uniforme) parmi $-30, -20, -10, 10, 20$ et 30 degrés.

On applique ensuite la procédure *VERIF* sur chaque élément de population.

Dans 1 cas sur 3, on choisit de générer une trajectoire non déviée afin que l'algorithme génétique puisse détecter rapidement les solutions sans conflit.

Beaucoup de choix sont empiriques et ne sont justifiés que qualitativement. Ces choix pourront dans des travaux ultérieurs être approfondis sur le plan quantitatif.

4.6.4 Fitness

La fonction fitness doit prendre en compte à la fois les contraintes de séparation des avions et la qualité de la résolution lorsque celle-ci est réalisée. Réduire à une seule valeur fitness tous les calculs effectués pour l'évaluation d'un individu est très réducteur et une grande partie de l'information utile est perdue dans des opérations d'additions (voir [DAN94]). Aussi, la fonction fitness unique sera remplacée par une matrice triangulaire inférieure F . Si le cluster à résoudre comprend n avions

FIGURE 4.21 – Surface occupée par la manœuvre

alors n sera la taille de la matrice F . Si $i \neq j$, alors $F_{i,j}$ mesure la gravité du conflit entre l'avion i et l'avion j . S'il n'y a pas de conflit, $F_{i,j} = 0$, $F_{i,j}$ croît avec la gravité du conflit. $F_{i,i}$ mesure la pénalisation de la trajectoire de l'avion i engendrée par ses manœuvres. Cette matrice fitness d'où on pourra extraire une fitness scalaire contient beaucoup plus d'information qu'une fitness scalaire et permet de définir les opérateurs de croisement et de mutation adaptés au problème de résolution de conflits.

Nous sommes face à un problème d'optimisation multi-critères. Nous avons retenu pour les applications les critères suivants :

- Le retard induit pour chaque avion doit être aussi faible que possible.
- Le nombre d'avions déviés et le nombre total de manœuvres doivent être aussi faible que possible, il se peut donc que les retards ne soient pas partagés par tous les avions.
- La durée d'une manœuvre doit être aussi faible que possible pour pouvoir rendre le plus tôt possible un avion disponible pour une autre manœuvre.
- Les trajectoires doivent respecter les normes de séparation.

Détaillons le calcul des éléments de la matrice F .

Calcul des éléments de la diagonale

Pour prendre en compte à la fois le retard dû à la manœuvre et la durée de la manœuvre elle-même, on peut mesurer la surface S occupée par la résolution (figure 4.21). Cette surface augmente à la fois avec le retard induit et la durée de la manœuvre. Il est intéressant de rappeler que cette notion de surface occupée est un des critères utilisés par les contrôleurs pour décider d'une manœuvre. Pour minimiser le nombre de manœuvres, on ajoute à S le nombre de manœuvres multiplié par un certain coefficient k . Pour un point tournant, il faut donc rajouter $2k$ à S , pour un offset, il faut rajouter $3k$ (pour les applications pratiques, k est fixé empiriquement à d^2).

$$F_{i,i} = S_i + k N_i$$

Evaluation des termes non diagonaux

A chaque pas de temps t , on évalue la différence (si elle est positive) $C_{t,i,j}$ entre la séparation standard et la distance entre les segments i et j représentant les

positions des avions i et j à t (voir partie 3.6). Ces valeurs sont additionnées pour tous les temps t pour donner $F_{i,j}$, mesure du conflit entre i et j .

$$F_{i,j} = \sum_{t=0}^{total\ time} (C_{t,i,j})$$

A partir de la matrice F , et pour les besoins de l'opérateur de sélection, on est en mesure de calculer une fitness f scalaire de la façon suivante :

$$\begin{aligned} \exists(i, j), F_{i,j} \neq 0 &\Rightarrow f = \frac{1}{2 + \sum_{i \neq j} F_{i,j}} \\ \forall(i, j), F_{i,j} = 0 &\Rightarrow f = \frac{1}{2} + \frac{1}{1 + \sum_i F_{i,i}} \end{aligned}$$

On remarquera que cette fitness permet de distinguer les individus pour lesquels il reste des conflits (leurs fitness sont inférieures à 0.5) des individus où il n'y a plus de conflit (leurs fitness sont supérieures à 0.5). Ce choix de fitness scalaire n'est pas le seul possible. Il serait intéressant dans un stade ultérieur de le discuter.

4.6.5 Opérateurs de croisement et de mutation adaptés

Pour les clusters de grande taille, il est intéressant d'utiliser la séparabilité partielle du problème et d'introduire l'opérateur de croisement adapté. Pour cela, à partir de la matrice fitness F , on peut définir pour l'avion ou le chromosome i sa fitness locale :

$$F_i = \sum_{j=1}^n (F_{i,j})$$

L'opérateur de croisement est décrit par la figure 4.22. Après avoir choisi deux parents A et B , on compare les fitness locales de leurs avions (ces fitness sont notées A_i et B_i sur la figure 4.22). Pour l'avion i , si $A_i \leq B_i - \delta$ (dans les applications $\delta = n \times d$) les deux enfants héritent de l'avion i du père A . Si $B_i \leq A_i - \delta$, les deux enfants héritent de l'avion i du père B . Si aucune de ces deux conditions n'est satisfaite, on applique l'opérateur de croisement barycentrique sur les avions i des parents A et B .

De même, on pourra définir un opérateur de mutation adapté décrit par la figure 4.23. On définit un seuil ϵ (dans les applications $\epsilon = d$). Après avoir choisi un individu à muter, on choisit un avion au hasard parmi ceux dont la fitness locale est supérieure à ϵ (sur la figure 4.23 l'avion 4 est choisi). Si l'on n'en trouve pas, on choisit un avion au hasard. Un des gènes de l'avion est choisi au hasard et modifié d'un pas de temps, ou de 10 degrés (s'il s'agit du dernier gène).

FIGURE 4.22 – Opérateur de croisement adapté

FIGURE 4.23 – Opérateur de mutation adapté

Les opérateurs de croisement et de mutation adaptés sont assez déterministes en début de convergence car il reste alors beaucoup de conflits et les écarts entre fitness locales des avions peuvent être très importants. Leur rôle est donc alors surtout de dégager une solution admissible (dont la fitness est supérieure à 0.5). Quand les solutions sans conflit deviennent suffisamment nombreuses, ces opérateurs deviennent moins déterministes (les fitness locales sont voisines). L'association de ces opérateurs avec une méthode de sharing permet d'éviter la convergence prématurée.

4.6.6 Sharing simplifié

On se propose d'utiliser une méthode de sharing simplifiée qui a l'avantage de distinguer de façon discrète deux solutions qui n'ont pas les mêmes caractéristiques. Ainsi, définissons la distance discrète suivante ($\alpha_i(A)$ est l'angle de déviation de l'avion i du chromosome A) :

- deux individus A et B sont séparés par une distance nulle si :

$$\forall i \in [1, n], \alpha_i(A) = \alpha_i(B)$$

- deux individus A et B sont séparés d'une distance 1 si :

$$\exists i \in [1, n], \alpha_i(A) \neq \alpha_i(B)$$

On dira que deux individus séparés par une distance nulle appartiennent à la même classe. Si l'on applique la méthode de sharing clusterisé avec $d_{min} < d_{max} < 1$, on remarque que cela revient à diviser la fitness de chaque individu par le nombre de représentants de sa classe.

4.6.7 Méthode locale en fin de convergence

Les algorithmes génétiques sont très efficaces pour résoudre des problèmes d'optimisation combinatoire de grande taille mais leur efficacité est moindre pour une recherche locale d'une bonne précision. En conséquence, il est très utile, après la dernière génération de l'algorithme génétique, d'appliquer une méthode d'optimisation locale sur la meilleure solution de chaque classe d'individus. Plusieurs méthodes locales peuvent être envisagées. Les méthodes les plus naturelles, pour la résolution de conflit, sont les méthodes ne requérant pas de calcul de gradient ou de Hessien. En effet, la fonction fitness est le résultat d'une simulation qui peut devenir coûteuse lorsque la taille du problème augmente.

Dans un premier temps, la méthode du simplexe (Nelder-Mead) [NM65, JKP72] a été programmée, moyennant quelques adaptations permettant de prendre en compte les contraintes gérées par la procédure *VERIF*. Les résultats obtenus avec cette méthode sont bons mais dans la plupart des exemples à n avions ($n > 10$) testés, le temps de calcul lié à cette méthode est très important (il peut atteindre plusieurs minutes).

Afin d'accélérer la convergence, une méthode locale simpliste a été envisagée : les gènes t_0 , t_1 et t_2 de tous les chromosomes de l'individu qu'on optimise sont successivement modifiés d'un pas de temps, de façon à améliorer la fitness globale sans générer de nouveaux conflits. Le processus s'arrête lorsque plus aucun gène ne peut être amélioré. Cette méthode permet, à chaque itération, de ne pas réévaluer tous les éléments de la matrice fitness, mais seulement les éléments de la ligne et de la colonne de la variable modifiée. Lorsque le nombre d'avions est important, le gain de temps peut devenir significatif.

On peut cependant craindre, en cas de mauvais conditionnement numérique du problème, que cette méthode soit largement inefficace. Le choix d'une méthode moins rudimentaire dépendra essentiellement du temps de calcul dont on dispose. Il sera essentiel d'étudier ce problème dans des travaux futurs.

4.6.8 Applications numériques

Les simulations précédentes ont été effectuées sur un pentium 133. Pour chaque simulation, nous précisons le temps moyen d'une optimisation de cluster. En fait, préciser le temps de calcul en secondes n'a qu'un intérêt qualitatif pour savoir si le problème est soluble dans l'état actuel de la technologie. Ce temps de calcul dépend pour beaucoup des moyens informatiques dont on dispose et sera probablement très réduit dans 2 ou 3 ans. Aussi, nous rajouterons le nombre d'appels à la fonction d'évaluation qui ne dépend que de la méthode utilisée.

La séparation standard choisie dans les applications est de $4 NM$ ($2 NM$ par avion). Lorsque rien n'est précisé, le sharing est utilisé et les avions volent à $400 kt$. Pour les représentations graphiques, l'unité utilisée est le soixantième de nautique¹⁰. Tous les exemples présentés sont fictifs.

Exemple montrant l'intérêt du modèle temps réel

Dans ce premier exemple, on considère un conflit très simple entre 2 avions (voir figure 4.24). Le temps d'anticipation Δ est fixé à 2 minutes et la vitesse des avions à $400 kt$ avec une incertitude de 5 pour cent. Dans ce cas, si l'avion 0 a une vitesse sol de $380 kt$ et l'avion 1 une vitesse sol de $420 kt$, il n'y aura finalement pas de conflit (la séparation standard est de $4 NM$). Au cas où les avions ne respecteraient pas leurs trajectoires, l'algorithme calcule des solutions dans le cas le plus défavorable. La figure 4.25 donne le résultat de l'algorithme génétique à $t = 3$ minutes, 5 minutes, 7 minutes, 9 minutes, 11 minutes et 13 minutes. Les traits épais représentent les deux minutes à venir qui ne peuvent être modifiées. Les traits fins décrivent les trajectoires optimales provisoires calculées par l'algorithme génétique. Les paramètres de l'algorithme génétique sont :

nombre de générations : 20

nombre d'individus : 50

10. Un soixantième de nautique représente une seconde d'arc à l'équateur.

FIGURE 4.24 – Conflit à deux avions

En raison de l'incertitude, la première optimisation donne des trajectoires robustes mais très pénalisantes. Au fur et à mesure que le temps passe, les trajectoires s'affinent car l'incertitude décroît. Finalement, à $t = 12$ le conflit disparaît. Le temps de calcul moyen d'une optimisation est de 10 secondes. L'AG nécessite 750 appels à la fonction d'évaluation. La méthode locale en fin de convergence requiert entre 40 et 50 appels à la fonction d'évaluation.

Exemple de problème à 5 avions

Dans cet exemple, 5 avions convergent vers le même point à une vitesse de 400 *kt* avec 3% d'erreur. Δ vaut 5 minutes. Les autres paramètres de l'algorithme génétique n'ont pas changé. La figure 4.26 donne les résultats. La distance minimale d'espacement des avions est de 6.01 *NM* malgré une forte incertitude sur les vitesses (3%). Le temps de calcul moyen d'une optimisation est de 20 secondes. L'AG nécessite 750 appels à la fonction d'évaluation. La méthode locale en fin de convergence requiert entre 25 et 150 appels à la fonction d'évaluation, avec une moyenne de 70 appels.

Le tableau 4.6 donne la distance minimale observée entre les avions pour différents pourcentages d'incertitudes et différents Δ . On observe, grâce à ce tableau, que lorsqu'on a simultanément une forte incertitude et que Δ est important, les avions s'évitent avec une marge importante. Ce résultat était prévisible. En effet, pour pouvoir rester efficace avec de fortes incertitudes sur les vitesses, il faut renouveler souvent l'optimisation.

Les résultats représentés ne donnent que la meilleure solution de ce problème à 5 avions. L'algorithme génétique propose grâce au sharing d'autres solutions proches de la meilleure solution (notamment la solution symétrique de la figure 4.26).

FIGURE 4.25 – $t=3, 5, 7, 9, 11, 13$ minutes

FIGURE 4.26 – $t=2, 6, 10, 14, 18, 22$ minutes

incertitude	0%	1%	2%	3%	4%	5%
1 min	4.00	4.24	4.30	4.55	4.65	4.70
2 min	4.03	4.35	4.52	5.00	5.30	5.77
3 min	4.01	4.44	4.89	5.16	5.61	5.97
4 min	4.02	4.50	5.02	5.63	6.10	6.68
5 min	4.03	4.57	4.89	6.01	6.61	7.32

TABLE 4.6 – Distance minimale entre les avions pour différentes valeurs de Δ .

Exemples de plusieurs solutions proches

Le but de cet exemple est de montrer que l'algorithme génétique est capable de générer plusieurs solutions proches de la meilleure solution trouvée, mais ayant des caractéristiques différentes. A partir de l'exemple 4.6.8, après la première optimisation, on a conservé toutes les solutions admissibles ayant une fitness au moins égale à 95% de la meilleure fitness. On obtient ainsi dans les mêmes conditions d'optimisation 13 configurations différentes. Six d'entre elles sont proposées figure 4.27.

On peut de même, sur un exemple plus simple (voir figure 4.28), observer les solutions proches de la meilleure solution trouvée proposées pour un conflit à trois avions. Le temps de calcul moyen d'une optimisation dans ce cas est de 15 secondes. L'AG nécessite 750 appels à la fonction d'évaluation. La méthode locale en fin de convergence requiert entre 10 et 110 appels à la fonction d'évaluation, avec une moyenne de 60 appels.

Efficacité des opérateurs adaptés

Pour tester l'efficacité des opérateurs de croisement et de mutation adaptés introduits en 4.22, nous proposons de résoudre deux conflits impliquant chacun 20 avions.

Le premier (figure 4.29) implique 20 avions convergeant simultanément sur un même point. Ce conflit a très peu de solutions car tous les avions sont en conflit deux à deux. En fait, deux solutions (au moins) existent : dans la première, tous les avions tournent à droite, dans la seconde, tous les avions tournent à gauche. La figure 4.30 donne le résultat de l'optimisation. Les paramètres de l'algorithme génétique sont les suivants :

nombre de générations : 20
éléments de population : 100

Le deuxième conflit (figure 4.31) regroupe 10 conflits à 2 avions indépendants. Dans ce cas, il existe de nombreuses solutions facilement identifiables à cause des

FIGURE 4.28 – Conflit à 3 avions, solutions proches de la meilleure solution observée.

FIGURE 4.29 – Conflit à 20 avions (convergeant sur un point)

symétries du problème. Cependant, l'algorithme génétique ne profite pas de ces symétries. La figure 4.32 donne le résultat de l'optimisation par algorithme génétique avec les mêmes paramètres que précédemment. Le temps de calcul moyen d'une optimisation est de 70 secondes. L'AG nécessite 1500 appels à la fonction d'évaluation. La méthode locale en fin de convergence requiert entre 50 et 800 appels à la fonction d'évaluation.

Pour étudier l'effet des opérateurs de croisement adaptés combinés à l'utilisation du sharing, 4 différents tests ont été effectués sur les deux conflits précédents en utilisant :

- les opérateurs adaptés sans sharing.
- les opérateurs adaptés avec sharing.
- les opérateurs classiques sans sharing.
- les opérateurs classiques avec sharing.

Pour ces quatre tests, nous avons mesuré pour chaque exemple la valeur du meilleur élément de population (figure 4.33 et 4.34) pour les 20 générations de l'algorithme génétique.

On peut d'abord constater que les opérateurs adaptés sont plus efficaces que les opérateurs classiques pour trouver des solutions admissibles. En effet, avec les opérateurs classiques, aucun des deux conflits n'est résolu avant la génération 20 (en fait, des tests montrent que la première solution admissible est obtenue après la génération 200). Avec les opérateurs adaptés, une solution admissible est toujours trouvée avant la génération 10 (Une solution est admissible si sa fitness est supérieure à 0.5).

Les figures 4.33 et 4.34 montrent que le sharing a tendance à freiner la crois-

FIGURE 4.31 – Conflit à 20 avions (10 conflits indépendants)

sance de la fitness du meilleur élément. Cependant la fitness finale est aussi bonne lorsqu'on utilise le sharing que lorsqu'on ne l'utilise pas. Le sharing permet d'obtenir dans le cas du conflit à 20 avions les deux solutions symétriques (tous les avions tournant à droite et tous les avions tournant à gauche).

Exemple de conflit de rattrapage

Le but de cet exemple est de montrer l'intérêt de la modélisation par offset dans le cadre d'un conflit en rattrapage. Dans l'exemple présenté, 5 avions se suivent sur la même route. L'avion de tête est le plus lent (350 *kt*), l'avion de queue est le plus rapide (550 *kt*). La population est fixée à 50 éléments. Le temps de calcul moyen d'une optimisation est de 20 secondes. L'AG nécessite 750 appels à la fonction d'évaluation. La méthode locale en fin de convergence requiert entre 10 et 140 appels à la fonction d'évaluation, avec une moyenne de 60 appels.

Les résultats sont représentés figure 4.35. On observe que l'offset est utilisé pour les 4 avions les plus rapides, le plus lent n'étant pas dévié. La déviation est d'autant plus importante que l'avion est rapide, ce qui reste en accord avec les conclusions de la partie 3.4.

Exemple de l'intérêt d'une résolution globale d'un cluster

Dans cet exemple, deux conflits indépendants à deux avions face à face sont regroupés dans le même cluster par l'intermédiaire d'un avion coupant leurs axes. La figure 4.36 donne le résultat de l'optimisation. On observe que la déviation de l'avion coupant les axes résoud 4 conflits (l'avion coupant les axes est en effet en conflit avec les 4 autres). Le temps de calcul moyen d'une optimisation est de 20 secondes. L'AG nécessite 750 appels à la fonction d'évaluation. La méthode locale

FIGURE 4.33 – Meilleure fitness en fonction de la génération courante (avions en cercle)

FIGURE 4.34 – Meilleure fitness en fonction de la génération courante (10 conflicts indépendants)

en fin de convergence requiert entre 10 et 210 appels à la fonction d'évaluation (en moyenne 50).

Exemple de cluster de grande taille sans symétrie apparente

Cet exemple a été fabriqué en faisant converger simultanément une douzaine d'avions vers une zone centrale d'environ 40×40 nautiques. Les résultats sont présentés figure 4.37. La population a été fixée à 100 éléments, le temps de calcul moyen d'une optimisation est de 50 secondes. L'AG nécessite 1500 appels à la fonction d'évaluation. La méthode locale d'optimisation requiert entre 30 (lorsque presque tous les avions sont contraints) et 780 (lorsque presque tous les avions sont libres) appels à la fonction d'évaluation.

4.6.9 Qualité des solutions obtenues

La qualité des solutions obtenues avec les algorithmes génétiques est difficile à évaluer pour plusieurs raisons :

- La structure de l'ensemble des solutions admissibles ne permet pas à un algorithme d'optimisation locale de trouver l'optimum global. Comme on peut l'observer dans l'annexe D, la structure de chaque composante connexe n'est pas convexe et l'optimum peut être "très isolé".
- La modélisation choisie pour tenir compte des contraintes opérationnelles ne permet pas d'envisager une fonction d'évaluation simple à calculer. Cette dernière devra être le résultat d'une simulation rendant très coûteuse l'évaluation d'un gradient, *a fortiori* d'un Hessien. Pour cette raison, nous n'avons pas envisagé d'utiliser un algorithme de type gradient sur le problème modélisé¹¹.

On peut envisager d'évaluer la qualité de la solution obtenue de la façon suivante : dans la partie 3.2.2, nous avons montré que la modélisation par offset permet de linéariser le problème d'optimisation. Cependant, pour un conflit à n avions, il faut résoudre $2^{\frac{n(n+1)}{2}}$ programmes linéaires. Dans [M94], F. Médioni fait calculer à un algorithme de simplexe [GM79] les 32768 programmes linéaires pour l'exemple à 5 avions convergeant sur un même point (voir figure 4.26). Pour cet exemple, la somme des retards de chaque avion est minimisée. On suppose que les avions sont déviés suffisamment tôt pour que les contraintes de séparation soient satisfaites pendant la mise en offset.

L'exemple a les caractéristiques suivantes : les 5 avions volent à 400 *kt* avec des caps de 270, 234, 198, 162 et 126 degrés. Au début de l'optimisation, ils sont à 100 *NM* du point de croisement de leurs trajectoires. L'angle de mise en offset est de 30 degrés.

11. L'investissement aurait été sans doute assez coûteux pour un résultat probablement inexploitable dès que l'on augmente la taille du problème

FIGURE 4.38 – Trajectoires d'évitement optimales pour un conflit à 5 avions : solutions symétriques.

Utilisation d'un simplexe sur le problème linéarisé

On obtient alors, grâce à l'algorithme de simplexe (répété 32768 fois), deux solutions optimales équivalentes :

- soit les quatre premiers avions sont déviés vers la droite de la même déviation, tandis que le cinquième n'est pas dévié.
- soit le premier avion n'est pas dévié, et les quatre suivants sont déviés vers la gauche de la même déviation.

Le temps de calcul est d'une centaine de secondes sur *HP720*. Ces deux solutions sont représentées sur la figure 4.38. Le retard total généré est de 167 secondes, ce qui représente 18.55 *NM* d'allongement total (4.63 *NM* par avion).

Utilisation de l'algorithme génétique

On peut simplifier l'algorithme génétique de façon à ne lui faire produire que des trajectoires d'offset, avec un angle fixe de 30 degrés et sans incertitude. On simplifie alors la fonction d'évaluation de façon à ce qu'elle ne prenne en compte que la somme des déviations, sans tenir compte du nombre de manœuvres effectuées. Les résultats des deux méthodes deviennent alors comparables.

Les paramètres utilisés sont les suivants :

Pas de temps : 12 s
Nombre de générations : 20
Taille de la population : 100

Les autres paramètres sont identiques à ceux utilisés dans les applications pré-

FIGURE 4.39 – Trajectoires obtenues par AG pour un conflit à 5 avions : meilleures solutions.

cédentes. Le temps de calcul est de l'ordre de 30 secondes. L'utilisation du sharing permet de conserver les meilleurs éléments de cluster dont la fitness dépasse 95% de la meilleure fitness. On obtient ainsi les deux solutions représentées figure 4.39. Les retards induits sont :

- $(43\ s, 45\ s, 43\ s, 45\ s, 0)$ pour le cas où le dernier avion n'est pas dévié et les autres tournent à droite.
- $(0, 45\ s, 43\ s, 45\ s, 43\ s)$ pour le cas où le premier avion n'est pas dévié et les autres tournent à gauche.

Le total est le même pour les deux solutions, à savoir 176 s, ce qui représente 19.55 NM d'allongement total.

Compte tenu du pas de temps de 12 s utilisé pour la fonction d'évaluation dans l'algorithme génétique, on peut considérer que les résultats obtenus par AG sont très bons. On peut considérer, sur cet exemple, que l'algorithme génétique permet d'obtenir les deux optima du problème.

Il est important de noter que si l'on avait voulu résoudre les 2097152 programmes linéaires du problème à 6 avions, il aurait fallu un peu plus de 3 heures sur HP720 (voir [M94]). L'intérêt des AGs devient alors évident.

4.6.10 Recherche des optima de Pareto

Dans ce paragraphe, on applique la méthode présentée dans la partie 4.4.7 au problème de résolution de conflits. L'exemple considéré est le même que précédemment. Le vecteur des critères à optimiser est tout simplement composé des fitness locales de chaque avion. On souhaite, à travers cet exemple, obtenir au moins 5 optima de Pareto, chacun ne déviant pas un des 5 avions. Pour cela, il a fallu prendre les paramètres suivants :

Nombre de générations : 100

Taille de la population : 200

Le temps de calcul est alors de l'ordre de 15 minutes sur pentium 133. Cette technique a été abandonnée car trop coûteuse en temps de calcul, mais mériterait sans doute d'être reconsidérée compte tenu de l'évolution rapide des puissances disponibles.

Conclusion

L'utilisation des algorithmes génétiques pour la résolution de conflits comporte de nombreux avantages :

- Ce sont des méthodes d'optimisation globale qui utilisent uniquement la fonction d'évaluation. Celle-ci peut être le résultat d'une simulation.
- Utilisés avec une méthode de sharing, ils permettent d'obtenir plusieurs optima locaux proches de la meilleure solution.

Les AGs se sont avérés très efficaces pour résoudre des conflits pouvant impliquer jusqu'à une vingtaine d'avions. L'introduction d'un opérateur de croisement adapté aux fonctions partiellement séparables combiné avec une méthode de sharing s'est toutefois révélée très utile pour obtenir plusieurs solutions différentes.

Cet opérateur permet de tenir compte de la structure particulière des fonctions partiellement séparables. Les expériences effectuées sur les différentes fonctions tests ont permis d'observer son efficacité. On peut toutefois construire des exemples pour lesquels l'opérateur favorise la convergence de l'algorithme vers un optimum local. Une étude théorique de cet opérateur reste à faire.

La recherche des optima de Pareto du problème multi-critères associé est très coûteuse et ne paraît pas envisageable pour l'instant dans un système en temps réel. Cette voie reste néanmoins à explorer.

L'utilisation d'une méthode locale en fin de convergence de l'AG permet d'améliorer les meilleurs éléments trouvés. Des travaux restent à faire sur le choix d'une méthode permettant de trouver un compromis acceptable entre efficacité et coût en temps de calcul.

La mise en œuvre des algorithmes génétiques reste cependant délicate car de nombreux paramètres doivent être fixés :

- La taille de la population et le nombre de générations.
- Les probabilités de croisement et de mutation.
- La génération de la population initiale.

- Les différents opérateurs de sélection, croisement, mutation, scaling et sharing.
- Le choix de δ pour le croisement adapté.

Dans un premier temps, on s'est attaché avant tout à tenter de résoudre des clusters d'avions de grandes tailles. Nous avons donc fixé beaucoup de paramètres sans justifier quantitativement tous les choix faits. Des mesures de sensibilité aux différents paramètres des AGs restent à faire.

Pour la résolution de conflits, il est difficile de comparer les AGs aux méthodes locales. La modélisation de l'incertitude sur les vitesses des avions et la discrétisation des angles de déviation ne permettent pas de calculer la fonction d'évaluation autrement que par une simulation. Les expérimentations faites avec LANCELOT ne prennent pas en compte ces éléments de modélisation. De plus, elles ne garantissent pas l'optimalité globale de la solution.

On a pu néanmoins, sur un exemple à 5 avions, comparer la qualité des résultats obtenus par algorithme génétique à ceux obtenus en linéarisant le problème et en utilisant une méthode de simplexe sur le problème linéarisé (la combinatoire restant forte). L'incertitude sur les vitesses des avions a été pour cela négligée, les manœuvres proposées restreintes à des offsets. Les résultats obtenus avec l'algorithme génétique sont très bons sur cet exemple.

Conclusion

Le travail effectué durant ces trois années a permis de dégager les difficultés mathématiques posées par le problème de résolution de conflits aériens en route. Pour un conflit à deux avions contraints en vitesse, la solution optimale ne peut pas se calculer analytiquement. La complexité de la structure de l'ensemble des solutions admissibles a été mise en évidence. Les tests pratiques utilisant un algorithme d'optimisation locale (LANCELOT) ont révélé leur impuissance à résoudre ce type de problème en un temps raisonnable.

Une part importante du travail fut consacrée à la modélisation du problème. Elle a permis de définir deux types de manoeuvres dans le plan horizontal permettant de tenir compte des contraintes liées au contexte aéronautique (présence du pilote dans la boucle, incertitude sur les vitesses des avions). La perte d'optimalité liée aux approximations point tournant et offset a été évaluée. Elle reste faible dès lors que le conflit est résolu suffisamment tôt. Un conflit à deux avions est d'autant plus coûteux à résoudre que l'angle d'incidence des trajectoires est faible et que le rapport des vitesses est proche de 1. Les possibilités de régulation en vitesse ont été mesurées, puis abandonnées. On retiendra l'importance de l'incertitude sur les vitesses : elle conditionne fortement l'instant de décision de manoeuvre, et de ce fait, la qualité de la résolution. On a finalement retenu dans le plan horizontal deux types de manoeuvres, point tournant et offset, qui permettent de résoudre la plupart des configurations à deux avions.

Parmi les méthodes d'optimisation globale, les algorithmes génétiques nous ont paru réunir un grand nombre d'avantages. Ne requérant pas d'hypothèse sur la fonction à optimiser, ils permettent d'obtenir non pas une solution optimale, mais plusieurs solutions différentes proches de la meilleure solution trouvée, offrant ainsi plusieurs options à un opérateur humain. Parmi les autres méthodes d'optimisation globale, le recuit simulé n'a pas été retenu car il ne garantit pas plus que les algorithmes génétiques l'optimalité de la solution, ne propose en fin de convergence qu'une seule solution et ne permet pas de tenir compte de la séparabilité partielle. Les méthodes d'optimisation par intervalles mériteraient sans doute que l'on s'y intéresse, même si l'on peut craindre que la complexité du calcul de la fonction d'évaluation rende leur utilisation coûteuse. D'autres méthodes globales exactes existent, requérant des propriétés que ne vérifient pas la fonction d'évaluation.

L'observation de la fonction à optimiser a pu montrer qu'elle était partiellement

séparable. Partant d'exemples simples de fonctions partiellement séparables pour lesquels les optima sont connus, nous avons créé un opérateur de croisement tenant compte de cette séparabilité partielle pour tenter d'améliorer la convergence des AGs. Cet opérateur adapté a ensuite été évalué sur d'autres fonctions tests, donnant des résultats encourageants. On a pu cependant fabriquer un exemple pour lequel l'opérateur adapté ne donne pas des résultats satisfaisants. Par ailleurs, la mise en œuvre de l'opérateur fait intervenir un paramètre δ qui joue un rôle important dans la convergence de l'algorithme. La sensibilité de la convergence de l'AG à ce paramètre reste à étudier. Les choix faits dans les applications ont été empiriques.

L'opérateur adapté a ensuite été testé sur le problème de résolution de conflits pour lequel il permet d'augmenter considérablement la taille des problèmes traités, tout en conservant un temps de calcul raisonnable. Il permet alors d'envisager une résolution en temps réel (plus exactement pseudo-réel, par exemple toutes les 4 ou 5 minutes).

La qualité des solutions obtenues par algorithme génétique a pu être évaluée sur un exemple à 5 avions, en réduisant les trajectoires à des offsets, et en annulant l'incertitude sur les vitesses. Les résultats obtenus avec l'algorithme génétique sont très bons sur cet exemple.

Par ailleurs, on a pu constater que la recherche des optima de Pareto du problème multi-critères associé au problème de résolution de conflits est très coûteuse, et ne paraît pas envisageable pour l'instant dans un système en temps réel. Cette voie reste néanmoins à explorer et pourrait s'avérer très utile en vue d'un système d'aide à la décision.

La mise en œuvre des algorithmes génétiques reste délicate car de nombreux paramètres doivent être fixés (taille de la population, nombre de générations, probabilités de croisement, de mutation, génération de la population initiale etc). Dans un premier temps, on s'est attaché à résoudre des clusters d'avions de grande taille sans remettre en cause les choix des paramètres quantitativement.

Un début d'intégration dans un simulateur de trafic est actuellement en cours. Les résultats obtenus à ce jour sont trop peu nombreux pour pouvoir être présentés dans cette thèse. Afin d'avoir un solveur réaliste, il faudra néanmoins envisager l'extension des résolutions à la dimension verticale. On peut alors s'attendre à ce que la structure de l'ensemble des solutions admissibles soit modifiée. Néanmoins, on préférera dans un premier temps ne modifier dans le plan vertical que les trajectoires des avions en montée ou en descente, les avions stables en croisière ne subissant que des déviations latérales. Le problème restera donc combinatoire à cause de ces derniers.

Les premiers tests effectués sur le simulateur réel ont montré que la taille des clusters d'avions à résoudre est très sensible aux incertitudes envisagées. Avec une incertitude de 1%, les clusters d'avions restent de taille raisonnable (inférieure à 20). Avec une incertitude de 7% dans le plan horizontal, et de 20% dans le plan vertical, le système sature rapidement. On a pu observer que la qualité des résolutions dépendait également de l'incertitude sur les vitesses. Un des enjeux principaux pour la conception des systèmes futurs sera sans doute l'amélioration de la

prévision de trajectoire.

Annexe A

Rappels mathématiques

Condition nécessaire pour les problèmes de Contrôle Optimal avec contraintes sur l'Etat

Plusieurs approches [JLS71, JWS63, E.K82, BH75] faisant apparaître différemment les conditions nécessaires d'optimalité pour des problèmes de Contrôle Optimal avec contraintes sur l'état reprennent les résultats théoriques rappelés ci-dessous.

On s'intéresse au problème de Contrôle Optimal suivant :

$$\dot{x}(t) = f(t, x(t), u(t)), \forall t \in [t_0, t_f] \quad (\text{A.1})$$

$$x(t_0) = x^0 \quad (\text{A.2})$$

$$h(x(t_f)) = 0 \quad (\text{A.3})$$

$$u(t) \in U, \forall t \in [t_0, t_f] \quad (\text{A.4})$$

$$S(t, x(t)) \leq 0, \forall t \in [t_0, t_f] \quad (\text{A.5})$$

$$\text{Min } J(x, u) = \int_{t_0}^{t_f} l(t, x(t), u(t)) dt \quad (\text{A.6})$$

où :

- t_0 et t_f sont fixés et positifs
- $x(t) \in \mathfrak{R}^n, u(t) \in \mathfrak{R}^m, U \subset \mathfrak{R}^m$
- $f : \mathfrak{R} \times \mathfrak{R}^n \times \mathfrak{R}^m \rightarrow \mathfrak{R}^n$
- $h : \mathfrak{R}^n \rightarrow \mathfrak{R}^{n_1}, S : \mathfrak{R} \times \mathfrak{R}^n \rightarrow \mathfrak{R}^{n_2}, l : \mathfrak{R} \times \mathfrak{R}^n \times \mathfrak{R}^m \rightarrow \mathfrak{R}$

avec f, h, S, l , suffisamment réguliers, par exemple au moins de classe C^1 sur leur domaine de définition.

Soit Ω l'ensemble des couples trajectoire commandes admissibles, c'est à dire tels que u est une fonction mesurable sur $[t_0, t_f]$ à valeurs dans U telle qu'il existe une fonction x absolument continue sur $[t_0, t_f]$ à valeurs dans \mathfrak{R}^n vérifiant (A.1), (A.2), (A.3) et (A.5).

On suppose que Ω n'est pas vide, on appelle alors solution (P) tout couple (\bar{x}, \bar{u}) tel que :

$$\forall (x, u) \in \Omega \quad J(\bar{x}, \bar{u}) \leq J(x, u)$$

On fera l'hypothèse que toute solution (\bar{x}, \bar{u}) de (P) est telle que \bar{u} est bornée (c'est à dire $\exists M > 0 |u(t)| \leq M, \forall t \in [t_0, t_f]$).

On dit que la contrainte sur l'état (A.5) est d'ordre p si la fonction $(t, \xi) \rightarrow S(t, \xi)$ est au moins p fois continuellement dérivable et si p est le premier entier tel que la fonction $S^{(k)}$ définie par :

$$(t, \xi, v) \rightarrow S^{(k)}(t, \xi, v) = \left(\frac{\partial S^{(k-1)}}{\partial \xi}(t, \xi) | f(t, \xi, v) \right) + \frac{\partial S}{\partial t}(t, \xi)$$

$$S^{(0)}(t, \xi, v) = S(t, \xi)$$

vérifie :

$$\frac{\partial S^{(k)}}{\partial v} = 0, \forall k \in \mathbb{N}, 1 \leq k \leq p-1$$

$$\frac{\partial S^{(p)}}{\partial v} \neq 0$$

L'entier p s'interprète comme l'ordre de la première des dérivées successives de S par rapport à t contenant explicitement la commande.

On suppose désormais que la contrainte est d'ordre $p = 1$. On associe au problème (P) les trois Hamiltoniens suivants :

– L'Hamiltonien "naturel", noté H , défini par :

$$H(t, x(t), u(t); \lambda_0, p(t)) = \lambda_0 l(t, x(t), u(t)) + (p(t) | f(t, x(t), u(t)))$$

où $(\cdot | \cdot)$ est le produit scalaire canonique dans \mathbb{R}^n , $\lambda_0 \in \mathbb{R}$ et $p : [t_0, t_f] \rightarrow \mathbb{R}^n$ désigne l'état adjoint.

– L'Hamiltonien "augmenté" d'ordre 0, noté \tilde{H} , défini par $\tilde{H}(t, x(t), u(t); \lambda_0, p(t), \mu)$ ou $\tilde{H}(t, x(t), u(t); \lambda_0, p(t), F)$ où λ_0 et p sont définis comme pour H , et μ est une mesure σ -définie sur $[t_0, t_f]$ dont la donnée équivaut à celle d'une fonction F à variation bornée "normalisée" sur $[t_0, t_f]$, c'est à dire une fonction à variation bornée, continue à gauche et telle que $F(0) = 0$. On pose :

$$\tilde{H}(t, x(t), u(t); \lambda_0, p(t), \mu) = \lambda_0 l(t, x(t), u(t)) + (p(t) | f(t, x(t), u(t))) + \int_{t_0}^{t_f} S(t, x(t)) d\mu(t)$$

où $\int_{t_0}^{t_f} S(t, x(t)) d\mu(t)$ s'écrit encore $\int_{t_0}^{t_f} S(t, x(t)) dF(t)$.

– L'Hamiltonien augmenté d'ordre 1, noté \tilde{H}_1 :

$$\tilde{H}_1(t, x(t), u(t); \lambda_0, p(t), \mu) = \lambda_0 l(t, x(t), u(t)) + (p(t) | f(t, x(t), u(t))) + \int_{t_0}^{t_f} S^{(1)}(t, x(t)) d\mu(t)$$

On a alors le théorème de Ioffé [Iof84] suivant :

Théorème A.1 – *Sous les données et hypothèses précédentes, si (\bar{x}, \bar{u}) est une solution de (P) , alors il existe un scalaire $\lambda_0 \geq 0$, un vecteur $\lambda_f \in \mathbb{R}^{n_2}$, une fonction $\bar{p} : [t_0, t_f] \rightarrow \mathbb{R}^n$ et une mesure μ positive σ -finie concentrée sur l'ensemble $\{t | \text{in}[t_0, t_f], S(t, x(t)) = 0\}$ non tous nuls tels que :*

– Pour tout $t \in [T_0, t_f]$:

$$p(t) = \int_t^{t_f} \frac{\partial H}{\partial x}(s, \bar{x}(s), \bar{u}(s); \lambda_0, \bar{p}(s)) ds + \int_t^{t_f} \frac{\partial S}{\partial x}(s, \bar{x}(s)) d\mu(\mathbf{A}) \quad (7)$$

–

$$p(t_f) - p(t_f - 0) = -\frac{\partial S}{\partial x}(t_f, \bar{x}(t_f)) \mu(t_f) - {}^t h'(x(t_f)) \cdot \lambda_f$$

– Pour presque tout $t \in [t_0, t_f]$:

$$H(t, \bar{x}(t), \bar{u}(t); \lambda_0, \bar{p}(t)) = \text{Min}_{v \in U} H(t, \bar{x}(t), v; \lambda_0, \bar{p}(t))$$

Ce théorème appelle les remarques suivantes :

1. On peut avoir $\lambda_0 = 0$, ce cas (dit “non régulier”) correspond à une non qualification des contraintes en (\bar{x}, \bar{u}) pour le problème d’optimisation en dimension infinie induit par le problème (P) . Il existe des Conditions suffisantes pour que l’on ait $\lambda_0 \neq 0$ (voir par exemple Maurer [Mau79]), mais leur vérification peut être difficile. Il est, en général, plus simple d’essayer de conclure sur chaque problème particulier. Si $\lambda_0 \neq 0$, on peut toujours choisir $\lambda_0 = 1$.
2. D’après le théorème de Radon-Nykodym, la mesure μ peut s’écrire $\mu = \mu_1 + \mu_2 + \mu_3$ où :
 - μ_1 est une mesure diffuse absolument continue par rapport à la mesure de Lebesgue, concentrée sur l’ensemble $T = \{t | \text{in}[t_0, t_f], S(t, x(t)) = 0\}$. Il lui correspond une fonction à variation bornée F_1 absolument continue sur $[t_0, t_f]$
 - μ_2 est une mesure atomique concentrée sur un ensemble dénombrable de points appartenant à T . Il lui correspond une fonction F_2 à variation bornée dite “fonction des sauts”.
 - μ_3 est une mesure diffuse, étrangère à la mesure de Lebesgue, concentrée sur un ensemble de points de mesure de Lebesgue nulle, contenu dans T . Il lui correspond une fonction F_3 à variation bornée et continue sur $[t_0, t_f]$.
 Si $\mu_2 \neq 0$ alors l’état adjoint peut présenter des discontinuités en des instants $\tau \in T$.
 Si $\mu_2 = 0$ et $\mu_3 = 0$, la fonction p est absolument continue sur $[t_0, t_f]$ et l’équation intégrale (A.7) se réduit à l’équation différentielle suivante :

$$\dot{p}(t) = -\frac{\partial H}{\partial x}(t, \bar{x}(t), \bar{u}(t); \lambda_0, \bar{p}(t)) - \frac{\partial S}{\partial x}(t, \bar{x}(t)) \cdot \bar{q}(t)$$

où :

$$\begin{aligned} \bar{q} &\in L^1([t_0, t_f], \mathfrak{R}^n), \bar{q} \geq 0, \forall t \in [t_0, t_f] \\ \bar{q}(t) \cdot S(t, x(t)) &= 0, \forall t \in [t_0, t_f] \end{aligned}$$

Si $\mu_2 = 0$ et $\mu_3 \neq 0$, la fonction p n'est pas absolument continue sur $[t_0, t_f]$, elle est continue et dérivable presque partout sur $[t_0, t_f]$ (mais bien qu'elle l'implique toujours, l'équation intégrale (A.7) ne se réduit pas à l'équation différentielle précédente).

A notre connaissance, on ne sait pas dans le cas général caractériser les cas $\mu_2 = 0$ et $\mu_3 = 0$ en terme de Contrôle Optimal. Aussi énonce-t-on les théorèmes suivants qui ne sont que des traductions du théorème précédent dans les cas où $\mu_3 = 0$ et où μ_2 ne possède qu'un nombre fini de masses ponctuelles (voir Hartl, Sethi, Vickson [HSV95]).

Théorème A.2 –

Sous les données du théorème précédent et si de plus, la mesure μ est telle que sa composante atomique est concentrée en un nombre fini de points de T , alors il existe un scalaire $\lambda_0 \geq 0$, un vecteur $\lambda_f \in \mathfrak{R}^{n_2}$, une fonction $\bar{p} : [t_0, t_f] \rightarrow \mathfrak{R}^n$ absolument continue par morceaux, continue à gauche sur $]t_0, t_f[$, une fonction $\bar{q} \in L^1([t_0, t_f], \mathfrak{R}^n)$, une fonction η définie sur $[t_0, t_f]$, nulle en dehors de l'ensemble des points de discontinuité de p , non tous nuls tels que :

1.

$$\begin{aligned} \dot{\bar{p}}(t) &= -\frac{\partial H}{\partial x}(t, \bar{x}(t), \bar{u}(t); \lambda_0, \bar{p}(t)) - \frac{\partial S}{\partial x}(t, \bar{x}(t)) \cdot \bar{q}(t) \\ p(t_f) - p(t_f - 0) &= -\frac{\partial S}{\partial x}(t_f, \bar{x}(t_f)) \eta(t_f) - {}^t h'(x(t_f)) \cdot \lambda_f \end{aligned}$$

2.

$$\begin{aligned} \bar{q} \geq 0 \quad \text{et} \quad \bar{q}(t) \cdot S(t, \bar{x}(t)) &= 0, \quad \forall t \in [t_0, t_f] \\ \bar{\eta} \geq 0 \quad \text{et} \quad \bar{\eta}(t) \cdot S(t, \bar{x}(t)) &= 0, \quad \forall t \in [t_0, t_f] \end{aligned}$$

3. Pour tout $t \in [t_0, t_f]$

$$H(t, \bar{x}(t), \bar{u}(t); \lambda_0, \bar{p}(t)) = \text{Min}_{v \in U} H(t, \bar{x}(t), v; \lambda_0, \bar{p}(t))$$

Il importe de remarque que l'état adjoint p postulé par ce théorème ne coïncide avec l'état adjoint postulé par le théorème précédent que si $\mu_3 = 0$.

Théorème A.3 – *Le théorème précédent est toujours valable si l'on remplace S par $S^{(1)}$.*

Annexe B

LANCELOT

Programmé en Fortran, LANCELOT¹ est un logiciel puissant d'optimisation locale sous contraintes non linéaires et a surtout servi d'outil pour étudier le problème de résolution de conflits sous diverses contraintes. Le programme en lui-même n'a pas été modifié. La description rapide qui suit montre que si LANCELOT permet de résoudre des problèmes qui ne rentrent pas dans les cadres classiques d'algorithme d'optimisation, sa complexité en fait un outil lourd à utiliser. L'optimisation fait appel à deux algorithmes imbriqués, *SBMIN* et *AUGLG* dont nous donnons une description rapide sans démonstration. Tous les détails et les références aux résultats théoriques impliqués dans LANCELOT sont cités dans [CGT92b, CGT92a].

B.1 *SBMIN*

SBMIN est un algorithme d'optimisation dans un pavé de \mathbb{R}^n . Si f est une fonction de \mathbb{R}^n dans \mathbb{R} deux fois continument différentiable, *SBMIN* permet de résoudre numériquement :

$$\min_{x \in \mathbb{R}^n} f(x) \quad (\text{B.1})$$

$$m_i \leq x_i \leq M_i, \quad 1 \leq i \leq n \quad (\text{B.2})$$

m_i et M_i peuvent prendre des valeurs infinies. *SBMIN* est une méthode itérative. Nous noterons $g(x)$ le vecteur des dérivées partielles de f au point x et $H(x)$ la matrice des dérivées secondes de f au point x . A la fin de l'itération k , un estimateur $x^{(k)}$ de la solution vérifiant les contraintes est donné. L'itération $k + 1$ a pour but de trouver un itéré $x^{(k+1)}$ vérifiant les contraintes qui soit sensiblement meilleur que $x^{(k)}$. Sous certaines hypothèses raisonnables, l'algorithme assure que tout point d'adhérence de la suite $(x^{(k)})$ converge vers un point stationnaire de f vérifiant les contraintes.

1. Large And Nonlinearly Constrained Extended Lagrangian Optimization Techniques

Dans l'itération $(k + 1)$, le modèle quadratique de la fonction objectif $f(x)$ est construit :

$$m^{(k)}(x) = f(x^{(k)}) + g(x^{(k)})^T(x - x^{(k)}) + \frac{1}{2}(x - x^{(k)})^T B^{(k)}(x - x^{(k)}) \quad (\text{B.3})$$

où $B^{(k)}$ est une approximation symétrique de $H(x^{(k)})$. On définit également un scalaire $\delta^{(k)}$, rayon d'une région de confiance :

$$\|x - x^{(k)}\| \leq \delta^{(k)}, \quad (\text{B.4})$$

dans laquelle on considère que $m^{(k)}(x)$ est une bonne approximation de $f(x)$. L'ordre de grandeur de $\delta^{(k)}$ est réévalué au cours de l'algorithme.

L'itération $k + 1$ comporte plusieurs étapes que l'on peut résumer comme suit :

1. Tester la convergence.
2. Trouver une approximation du point de Cauchy généralisé du modèle quadratique respectant les contraintes et dans la région de confiance.
3. Trouver un nouveau point dans la région de confiance qui réduise le modèle quadratique et respecte les contraintes.
4. Tester l'adaptation du modèle à la fonction au nouveau point. Si elle est bonne, garder ce point comme nouvel itéré, sinon, garder l'ancien point et ajuster le rayon de la région de confiance.

Nous allons maintenant détailler les étapes.

1. Les conditions du premier ordre pour qu'un point x^* soit solution du problème (B.1) (B.2) sont que le gradient projeté s'annule à x^* . Le gradient projeté de $f(x)$ dans le pavé admissible (B.2) est :

$$x - P(x - g(x), m, M)$$

$P(x, m, M)$ est défini par :

$$\begin{aligned} P(x, m, M)_i &= m_i, \quad x_i < m_i \\ &= M_i, \quad x_i > M_i \\ &= x_i, \quad m_i \leq x_i \leq M_i \end{aligned}$$

La convergence de la méthode peut être mesurée par la valeur du gradient projeté à $x^{(k)}$. L'algorithme sera arrêté si :

$$\|x^{(k)} - P(x^{(k)} - g(x^{(k)}), m, M)\| \leq \epsilon_g$$

ϵ_g est la précision que l'on choisit d'obtenir. D'autres critères d'arrêts que nous ne détaillerons pas ici existent.

2. Le point de Cauchy généralisé.

La minimisation approchée du modèle quadratique (B.3) avec le pavé admissible (B.2) et la région de confiance (B.4) se fait en deux étapes. Dans la première, une approximation du point de Cauchy généralisé est évaluée. Ce point est important pour deux raisons. D'abord, la convergence de l'algorithme vers un point où le gradient projeté est nul peut être garantie pourvu que la valeur du modèle quadratique à la fin de l'itération ne dépasse pas celle au point de Cauchy généralisé. Ensuite, l'ensemble des variables valant leur borne au point de Cauchy généralisé au cours de l'algorithme donne souvent une excellente prédiction des variables valant leur borne pour la solution finale. Déterminer une valeur approchée de ce point est, de ce fait, important.

Soit $D^{(k)}$ une matrice diagonale définie positive et soit $G^{(k)} = D^{(k)} g(x^{(k)})$. L'arc de Cauchy est défini par :

$$x^{(k)}(t) = P(x^{(k)} - tG^{(k)}, l, u)$$

pour tout $t \geq 0$. Parcourant l'arc, t croissant, le point de Cauchy généralisé est le premier minimum local de $m^{(k)}(x^{(k)}(t))$, à l'intérieur de la région de confiance (B.4). Il existe plusieurs façons d'estimer le point de Cauchy généralisé que nous ne détaillerons pas ici.

3. Déterminer une bonne approximation du point de Cauchy généralisé assure la convergence de l'algorithme. Il faut maintenant s'assurer que cette convergence s'effectue à une vitesse raisonnable. Ceci est assuré, si nécessaire par une réduction du modèle quadratique. Les variables du point de Cauchy généralisé valant leur borne sont fixées. On tente ensuite de réduire le modèle quadratique en changeant les valeurs des variables restées libres. Ceci peut se faire de diverses façons. L'une d'entre elles est détaillée dans [CGT92b].
4. Un point $x^{(k)}$ qui réduit suffisamment le modèle quadratique est trouvé à cette étape. Or on souhaite réduire la fonction objectif $f(x)$ et non le modèle quadratique. Le succès de l'itération k peut être mesuré par le rapport entre la réduction de la fonction objectif et la réduction du modèle quadratique :

$$\rho^{(k)} = \frac{f(x^{(k)}) - f(x^{(k,j)})}{m^{(k)}(x^{(k)}) - m^{(k)}(x^{(k,j)})}$$

Si ce rapport est négatif ou petit devant 1, l'itération est un échec. Par contre si le ratio est grand, l'itération est un succès. Soit $0 < \mu < 1$. On choisira de mettre à jour $x^{(k+1)}$ de la façon suivante :

$$\begin{aligned} x^{(k+1)} &= x^{(k,j)}, \text{ si } \rho^{(k)} > \mu \\ &= x^{(k)}, \text{ sinon} \end{aligned}$$

On remarquera que puisque $\mu > 0$, il se peut que l'on refuse le meilleur point rencontré. On pourrait palier à cet inconvénient en stockant ce point et en le comparant au meilleur point à la fin de SBMIN.

Il faut également réduire le rayon de la région de confiance. Si ρ est négatif ou petit, le modèle n'a pas donné une bonne représentation de la fonction objectif, alors que si le rayon est proche de un, la représentation de la fonction objectif est bonne. Dans le premier cas, le rayon de la région de confiance doit être réduit. Dans le deuxième cas, il est raisonnable de penser qu'il peut être augmenté. Soit $\mu < \nu < 1$ et $0 < \gamma_0 < 1 \leq \gamma_2$. Le rayon de la région de confiance est mis à jour de la façon suivante :

$$\begin{aligned}\delta^{(k+1)} &= \gamma_0^{(k)} \delta^{(k)}, \text{ si } \rho^{(k)} \leq \mu \\ &= \delta^{(k)}, \text{ si } \mu < \rho^{(k)} \leq \nu \\ &= \gamma_2^{(k)} \delta^{(k)}, \text{ sinon}\end{aligned}$$

où $\gamma_0^{(k)} \in [\gamma_0, 1)$ et $\gamma_2^{(k)} \in [1, \gamma_2]$. D'autres détails sont donnés dans [CGT92b]. Si l'itéré a changé, on recalcule le gradient et le Hessien approché de la fonction objectif.

B.2 AUGLG

AUGLG permet de résoudre le problème de minimisation sous contraintes générales suivantes :

$$\min_{x \in \mathbb{R}^n} f(x) \tag{B.5}$$

$$m_i \leq x_i \leq M_i, \quad 1 \leq i \leq n \tag{B.6}$$

$$c_j(x) = 0, \quad 1 \leq j \leq m \tag{B.7}$$

où les contraintes égalités ne sont pas nécessairement linéaires. On suppose ici que f et c_j sont deux fois continuellement différentiables et que toute borne de (B.6) peut être infinie. On notera $a_j(x)$ le vecteur des dérivées partielles $\nabla_x c_j(x)$ et $H_j(x)$ la matrice du Hessien $\nabla_{xx} c_j(x)$. Pour traiter des contraintes inégalités, il suffit de rajouter une variable supplémentaire dans l'inégalité et de borner cette variable dans les équations (B.6).

AUGLG utilise SBMIN de façon répétée. La fonction objectif et les contraintes sont introduites dans la fonction Lagrangien augmenté :

$$\phi(x, \lambda, S, \mu) = f(x) + \sum_{i=1}^m \lambda_i c_i(x) + \frac{1}{2\mu} \sum_{i=1}^m s_{ii} c_i(x)^2$$

où les composantes λ_i du vecteur λ sont les estimateurs des multiplicateurs de Lagrange, les valeurs s_{ii} de la matrice diagonale S sont des variables d'échelle positifs et μ est appelé paramètre de pénalisation.

Le problème de minimisation sous contrainte (B.5)-(B.7) est résolu en trouvant un minimiseur approché de ϕ pour une séquence appropriée d'estimateurs des multiplicateurs de Lagrange, de variables d'échelle et de paramètres de pénalisation.

L'itération $k+1$ de *AUGLG* se fait en trois étapes. Au début de l'itération, des estimateurs des multiplicateurs de Lagrange $\lambda^{(k)}$, des variables d'échelle $S^{(k)}$, et un paramètre de pénalisation $\mu^{(k)}$ sont donnés. Les trois étapes sont les suivantes :

1. La convergence d'*AUGLG* est tout d'abord testée.
2. *SBMIN* est utilisé pour déterminer une approximation $x^{(k+1)}$ de la fonction Lagrangien augmenté $\phi(x, \lambda^{(k)}, S^{(k)}, \mu^{(k)})$ avec le pavé admissible (B.6).
3. On met à jour les estimateurs des multiplicateurs de Lagrange, les variables d'échelle, le paramètre de pénalisation et les tolérances de convergence.

Nous allons maintenant détailler quelque peu ces trois étapes.

1. Les conditions du premier ordre pour qu'un point admissible x^* soit solution du problème (B.5)-(B.7) requièrent qu'il existe des multiplicateurs de Lagrange λ^* pour lesquels le gradient projeté de la fonction de Lagrange en x^* et λ^* et les contraintes (B.7) s'annulent. La fonction Lagrangien est :

$$L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i c_i(x)$$

La convergence d'*AUGLG* est mesurée par la taille du gradient projeté et des contraintes à $x^{(k)}$ et $\lambda^{(k)}$. On arrêtera donc l'algorithme si :

$$\begin{aligned} \|x^{(k)} - P(x^{(k)} - \nabla_x L(x^{(k)}, \lambda^{(k)}), m, M)\| &\leq \epsilon_l \\ &\text{et} \\ \|c(x^{(k)})\| &\leq \epsilon_c \end{aligned}$$

pour des valeurs de ϵ_l et ϵ_c suffisamment petites.

2. La convergence du Lagrangien augmenté est garantie sous des hypothèses très faibles, si le paramètre de pénalisation est réduit progressivement à 0, même sans considérer la valeur des estimateurs des multiplicateurs de Lagrange. Ceux-ci peuvent même diverger pourvu que $\mu^{(k)} \|\lambda^{(k)}\|$ converge vers zero. Cependant, il est plus difficile d'optimiser (B.8) quand $\mu^{(k)}$ est petit. Heureusement, un choix judicieux des estimateurs de multiplicateurs de Lagrange assure également la convergence à μ fixé pourvu que $x^{(k)}$ soit proche de x^* . Ainsi, on autorise $\mu^{(k)}$ à décroître jusqu'à ce que l'on soit sûr que $x^{(k)}$ est proche de x^* . $\mu^{(k)}$ est alors fixé et les estimateurs des multiplicateurs de Lagrange sont ensuite ajustés de manière à terminer la convergence. On peut mesurer si l'on est dans un voisinage de x^* en contrôlant la décroissance de $\|c(x^{(k)})\|$. A chaque itération, on sort de *SBMIN* si :

$$\|x^{(k+1)} - P(x^{(k+1)} - \nabla_x \phi(x^{(k+1)}, \lambda^{(k)}, S^{(k)}, \mu^{(k)}), m, M)\| \leq \omega^{(k)}$$

est satisfait pour une certaine valeur de $\omega^{(k)}$. On teste ensuite si :

$$\|c(x^{(k+1)})\| \leq \eta^{(k)}$$

pour un certain seuil $\eta^{(k)}$. Si c'est le cas, le paramètre de pénalisation reste inchangé et les estimateurs des paramètres de Lagrange sont mis à jour. Sinon, le paramètre de pénalisation est réduit sans changer les estimateurs des multiplicateurs de Lagrange.

3. Les paramètres $\mu^{(k)}$, $\lambda^{(k)}$, $\omega^{(k)}$ et $\eta^{(k)}$ peuvent être mis à jour de sorte que la convergence d'*AUGLG* soit assurée en gardant le paramètre de pénalisation supérieur à un réel strictement positif. Les seuils μ_0 , ω_0 , η_0 , τ , α_ω , β_ω , α_η , et β_η satisfaisant :

$$\begin{aligned}\mu_0 &< 1 \\ \alpha_\eta &< \min(1, \alpha_\omega) \\ \beta_\eta &< \min(1, \beta_\omega)\end{aligned}$$

Dans *AUGLG*, on choisit $\tau = 0.1$, $\alpha_\omega = 1$, $\alpha_\eta = 0.1$ et $\beta_\eta = 0.9$. On pose :

$$\begin{aligned}\mu^{(0)} &= \mu_0 \\ \omega^{(0)} &= \omega_0(\mu^{(0)})^{\alpha_\omega} \\ \eta^{(0)} &= \eta_0(\mu^{(0)})^{\alpha_\eta}\end{aligned}$$

Sauf spécification contraire, *AUGLG* fixe $\lambda^{(0)} = 0$ et $S^{(0)} = I$.

Si (B.8) est satisfaite :

$$\lambda^{(k+1)} = \lambda^{(k)} + S^{(k)} \frac{c(x^{(k)})}{\mu^{(k)}}$$

Les autres paramètres sont fixés comme suit :

$$\begin{aligned}\mu^{(k+1)} &= \mu^{(k)} \\ \omega^{(k+1)} &= \omega^{(k)}(\mu^{(k+1)})^{\beta_\omega} \\ \eta^{(k+1)} &= \eta^{(k)}(\mu^{(k+1)})^{\beta_\eta}\end{aligned}$$

Si (B.8) n'est pas satisfaite :

$$\begin{aligned}\mu^{(k+1)} &= \tau \mu^{(k)} \\ \omega^{(k+1)} &= \omega_0(\mu^{(k+1)})^{\alpha_\omega} \\ \eta^{(k+1)} &= \eta_0(\mu^{(k+1)})^{\alpha_\eta} \\ \lambda^{(k+1)} &= \lambda^{(k)}\end{aligned}$$

Le choix du paramètre de pénalisation initial est difficile à justifier. Par défaut, *AUGLG* fixe $\mu_0 = 0.1$.

Annexe C

Comparaison point tournant - offset

Les figures suivantes sont citées dans la partie 3 mais ont été mises en annexes pour ne pas charger cette partie.

FIGURE C.1 – Allongements x_- et x_{t-} en fonction de t (variant de 0.5 à 5) et ϕ (variant de 0 à Π), l'unité de temps est le temps mis par les avions pour parcourir une séparation standard.

FIGURE C.2 – Allongements x_+ et x_{t+} en fonction de t (variant de 0.5 à 5) et ϕ (variant de 0 à Π), l'unité de temps est le temps mis par les avions pour parcourir une séparation standard.

FIGURE C.3 – Augmentation de l’allongement $x_{t-} - x_-$ en fonction de t (variant de 0.5 à 5) et ϕ (variant de 0 à Π), l’unité de temps est le temps mis par les avions pour parcourir une séparation standard.

FIGURE C.4 – Comparaison des allongements en % offset et point tournant, en fonction du temps d’anticipation (l’unité de temps est le temps mis par les avions pour parcourir une séparation standard), l’avion mobile passant derrière (à gauche), ou devant (à droite). $\phi = \frac{\pi}{2}$.

FIGURE C.5 – Comparaison des allongements en % offset et point tournant, en fonction du temps d'anticipation (l'unité de temps est le temps mis par les avions pour parcourir une séparation standard), l'avion mobile passant derrière (à gauche), ou devant (à droite). $\phi = \frac{\pi}{2}$.

Annexe D

Point tournant : étude dans \mathfrak{R}^4

L'étude suivante a pour but de donner une représentation de la forme de l'espace des points tournants admissibles dans \mathfrak{R}^4 . Nous savons d'ores et déjà que cet espace est composé de deux composantes connexes, l'avion 1 pouvant passer devant ou derrière l'avion 2. Compte tenu des difficultés de représentation dans \mathfrak{R}^4 , on peut fixer un des deux points tournants, et observer l'espace des points tournants admissibles dans \mathfrak{R}^2 cette fois.

D.1 Calcul des solutions admissibles et de leur fonction coût

Supposons qu'à t_1 l'avion a ait pour coordonnées a_1 et a_2 , et qu'à t_2 , ses coordonnées soient b_1 et b_2 . Supposons de même pour l'avion c avec c et d à la place de a et b . Enfin, on suppose que les avions a et c ne sont pas en évolution entre t_1 et t_2 .

Posons :

$$\begin{aligned}ac_1 &= c_1 - a_1 \\ac_2 &= c_2 - a_2 \\bd_1 &= d_1 - b_1 \\bd_2 &= d_2 - b_2 \\q_1 &= ac_1 - bd_1 \\q_2 &= ac_2 - bd_2 \\det &= q_1 \cdot bd_2 - q_2 \cdot bd_1\end{aligned}$$

On a montré alors que les avions restent séparés de d sur le segment si et seulement si :

- Si $q_1 \cdot bd_1 + q_2 \cdot bd_2 \geq 0$, $(bd_1)^2 + (bd_2)^2 \geq d^2$.
- Si $q_1 \cdot ac_1 + q_2 \cdot ac_2 \leq 0$, $(ac_1)^2 + (ac_2)^2 \geq d^2$.
- Sinon, $(det)^2 \geq d^2((q_1)^2 + (q_2)^2)$

Par ailleurs, la fonction coût associée à deux points tournants peut se calculer très facilement, il suffit pour cela de déterminer la somme des allongements des trajectoires de chaque avion induite par les points tournants et de la rapporter à la longueur initiale des trajectoires. On s'intéressera par la suite à d'autres types de fonctions coût, mais pour l'illustration de notre propos, ce choix paraît suffisant. Partant des fonctions d'admissibilité et de coût très simples précédentes, on peut construire un algorithme qui nous calcule les solutions admissibles et optimales pour le problème à deux avions et deux points tournants. Il suffit, en effet de discrétiser l'espace de recherche et de le balayer en partant du centre vers l'extérieur, de façon à ne pas perdre trop de temps. Néanmoins, cela reste coûteux pour un problème assez simple à résoudre. C'est pourquoi, sans tenter d'étendre la méthode à plus de deux avions, nous nous servirons de cet algorithme pour illustrer graphiquement la complexité de notre espace de recherche dans \mathbb{R}^4 .

D.2 Recherche du coût minimum global

Pour un point tournant fixé, on peut s'intéresser au coût total minimal pour les deux avions lorsque le deuxième point tournant varie. On peut alors représenter ce coût en fonction du point tournant fixé par un graphique en $3D$. Pour deux avions en conflit à angle droit, on observe la figure D.1, pour laquelle l'angle d'incidence des deux avions vaut $\frac{\pi}{2}$, les avions volent à 400 kt pendant 15 min et la séparation standard vaut 10 NM (Le temps d'anticipation est donc de 5 fois le temps mis pour parcourir une séparation standard).

On observe la présence de deux optima locaux. On observe également la présence d'un point selle entre ces deux optima, qui correspond au cas où l'avion 1 n'est pas dévié.

D.3 Représentation de l'espace des solutions admissibles

On peut représenter, lorsque la trajectoire d'un des deux avions est fixée, l'espace des points tournants admissibles pour l'autre avion, ainsi que l'allongement total engendré par ces points tournants. Cet exercice nous permet d'observer visuellement la forme du sous-ensemble de \mathbb{R}^2 dans lequel on recherche nos solutions, ainsi que la forme de la fonction à minimiser sur cet espace. Il est clair que la fonction à minimiser que nous avons choisie, à savoir la somme des allongements des deux trajectoires est convexe sur \mathbb{R}^2 , son minimum étant atteint pour un point tournant situé sur le segment reliant l'origine à la destination de l'avion observé. Toute la difficulté du problème vient du fait que l'espace des solutions admissibles n'est ni convexe, ni même connexe, et c'est en cela que ces représentations sont intéressantes. Les deux composantes connexes de l'espace des fonctions admissibles apparaissent alors clairement. Il est intéressant d'observer la forme de ces composantes connexes, et la position des optima sur celles-ci.

Dans les représentations graphiques D.2, D.3, D.4, D.5, D.6, D.7, D.8, D.9,

FIGURE D.1 – Inverse du coût global en fonction de la position de l'avion 1, l'avion 2 n'étant pas dévié. Les avions 1 et 2 vont respectivement de O_1 vers D_1 et de O_2 vers D_2 . $v_1 = v_2 = 400 kt$, la séparation standard vaut $10 NM$, l'anticipation vaut 5 séparations standard.

Dans un premier temps, on peut s'intéresser aux conflits entre deux avions dont la vitesse est identique et faire varier l'angle d'incidence. On peut ainsi observer l'évolution de l'ensemble des solutions admissibles pour chacun des deux avions lorsque l'autre est fixé et lorsque l'incidence des deux avions varie de Π à 0, (voir figures D.2, D.3, D.4, D.5, D.6, D.7, D.8, D.9). Dans les exemples choisis, les avions volent à 400 *kt* pendant 15 *min* avec une séparation standard de 10 *NM*. Le temps d'anticipation vaut donc 5 fois le temps mis pour parcourir une séparation standard. Les avions 1 et 2 volent respectivement de O_1 vers D_1 et de O_2 vers D_2 . Le point tournant de l'avion fixé est choisi de telle sorte qu'à l'optimum pour l'avion libre, le minimum d'allongement global soit atteint pour les deux avions. On observe bien la présence de deux composantes connexes lorsque les deux avions sont à des angles d'incidence forte, mais il est intéressant d'observer la localisation du minimum lorsque l'angle d'incidence diminue. En effet, après la déformation du "canyon" et l'apparition d'une "presqu'île", on observe la formation d'une île de plus en plus réduite. On imagine alors la difficulté pour un algorithme de gradient classique d'arriver au minimum global d'allongement.

On peut, pour citer un autre exemple caractéristique, s'intéresser au cas de deux avions suivant la même route, l'avion 2 allant deux fois plus lentement que l'avion 1. On observe alors également la forme de l'espace des solutions admissibles lorsque l'on ne bouge que l'avion le plus rapide (voir figure D.10) ou lorsque l'on ne bouge que le plus lent (voir figure D.11), l'anticipation valant cinq fois la norme de séparation. Dans le premier cas, on observe quatre composantes connexes, qui correspondent pour l'avion le plus rapide à un passage soit devant, soit derrière, soit à droite, soit à gauche de l'avion le plus lent. Dans le deuxième cas, on observe deux composantes connexes symétriques. Il est à noter que le plus faible allongement est obtenu lorsque l'on bouge l'avion le plus lent, (On remarquera que dans la partie 3.4, la modélisation par offset donne un résultat inverse).

D.3. REPRÉSENTATION DE L'ESPACE DES SOLUTIONS ADMISSIBLES 187

FIGURE D.2 – Points tournants admissibles pour l'avion 1, angle d'incidence : 180 degrés, vitesses égales, allongement minimum : 0.8%.

FIGURE D.3 – Points tournants admissibles pour l'avion 2, angle d'incidence : 180 degrés, vitesses égales, allongement minimum : 0.8%.

FIGURE D.4 – Points tournants admissibles pour l'avion 1, angle d'incidence : 90 degrés, vitesses égales, allongement minimum : 1.2%.

FIGURE D.5 – Points tournants admissibles pour l'avion 2, angle d'incidence : 90 degrés, vitesses égales, allongement minimum : 1.2%.

*D.3. REPRÉSENTATION DE L'ESPACE DES SOLUTIONS ADMISSIBLES*189

FIGURE D.6 – Points tournants admissibles pour l'avion 1, angle d'incidence : 54 degrés, vitesses égales, allongement minimum : 2.3%.

FIGURE D.7 – Points tournants admissibles pour l'avion 2, angle d'incidence : 54 degrés, vitesses égales, allongement minimum : 2.3%.

FIGURE D.8 – Points tournants admissibles pour l'avion 1, angle d'incidence : 18 degrés, vitesses égales, allongement minimum : 4.9%.

FIGURE D.9 – Points tournants admissibles pour l'avion 2, angle d'incidence : 18 degrés, vitesses égales, allongement minimum : 4.9%.

*D.3. REPRÉSENTATION DE L'ESPACE DES SOLUTIONS ADMISSIBLES*191

FIGURE D.10 – Points tournants admissibles pour l'avion 1, angle d'incidence nul, vitesse de l'avion 2 moitié , allongement minimum : 0.8%.

FIGURE D.11 – Points tournants admissibles pour l'avion 2, angle d'incidence nul, vitesse de l'avion 2 moitié , allongement minimum : 1.7%.

Annexe E

AG : résultats théoriques

E.1 Les algorithmes génétiques à codage binaire

Historiquement, les algorithmes génétiques binaires sont les plus anciens et ont donc été les plus étudiés sur le plan théorique. Goldberg [Gol89a] a largement développé cette théorie résumée dans [AS92].

E.1.1 Définitions fondamentales

Définition E.1 – Séquence – On appelle séquence A de longueur $l(A)$ une suite $A = a_1 a_2 \dots a_l$ avec $\forall i \in [1, l], a_i \in V = \{0, 1\}$.

En codage binaire, les chromosomes sont des séquences.

Définition E.2 – Schéma – On appelle schéma H de longueur l une suite $H = a_1 a_2 \dots a_l$ avec $\forall i \in [1, l], a_i \in V^+ = \{0, 1, *\}$.

Une $*$ en position i signifie que a_i peut être indifféremment 0 ou 1.

Définition E.3 – Instance – Une séquence $A = a_1 \dots a_l$ est une instance d'un schéma $H = b_1 \dots b_l$ si pour tout i tel que $b_i \neq *$ on a $a_i = b_i$.

Ainsi, $H = 010 * 0101$ est un schéma et les séquences 01000101 et 01010101 sont des instances de H (ce sont même les seules).

Définition E.4 – Position fixe, position libre – Soit un schéma H . On dit que i est une position fixe de H si $a_i = 1$ ou $a_i = 0$. On dit que i est une position libre de H si $a_i = *$.

Définition E.5 – Ordre d'un schéma – On appelle ordre du schéma H le nombre de positions fixes de H . On note l'ordre de H $o(H)$.

Par exemple, le schéma $H = 01 * * 10 * 1$ a pour ordre $o(H) = 5$, le schéma $H' = * * * * * 101$ a pour ordre $o(H') = 3$. On remarquera qu'un schéma H de longueur $l(H)$ et d'ordre $o(H)$ admet $2^{(l(H)-o(H))}$ instances différentes.

Définition E.6 – Longueur fondamentale – On appelle longueur fondamentale du schéma H la distance séparant la première position fixe de H de la dernière position fixe de H . On note cette longueur fondamentale $\delta(H)$.

Ainsi, le schéma $H = 1 * * 01 * * *$ a pour longueur fondamentale $\delta(H) = 5 - 1 = 4$, le schéma $H' = 1 * * * * * 1$ a pour longueur fondamentale $\delta(H') = 8 - 1 = 7$, et pour le schéma $H'' = * * 1 * * * *$ nous avons $\delta(H'') = 3 - 3 = 0$.

Définition E.7 – Adaptation d'une séquence – On appelle adaptation d'une séquence A une valeur positive que nous noterons $f(A)$.

f est la fonction objectif ou fitness du problème à résoudre.

Définition E.8 – Adaptation d'un schéma – On appelle adaptation d'un schéma H la valeur

$$f(H) = \frac{\sum_{i=1}^{2^{(l(H)-o(H))}} f(A_i)}{2^{(l(H)-o(H))}}$$

où les A_i décrivent l'ensemble des instances de H , c'est-à-dire la moyenne des adaptations de ses instances.

E.1.2 Effets de la reproduction

Soit un ensemble $S = \{A_1, \dots, A_i, \dots, A_n\}$ de n séquences de bits tirées aléatoirement. Durant la reproduction, chaque séquence A_i est reproduite avec une probabilité :

$$p_i = \frac{f(A_i)}{\sum_{i=1}^n f(A_i)}$$

Supposons qu'il y ait à l'instant t un nombre $m(H, t)$ de séquences représentant le schéma H dans la population S . A l'instant $t+1$, statistiquement, ce nombre vaut :

$$m(H, t+1) = m(H, t) \cdot n \cdot \frac{f(H)}{\sum_{i=1}^n f(A_i)}$$

Posons

$$\bar{f}_t = \frac{\sum_{i=1}^n f(A_i)}{n}$$

\bar{f}_t représente la moyenne de l'adaptation des séquences à l'instant t . La formule précédente devient :

$$m(H, t+1) = m(H, t) \frac{f(H)}{\bar{f}_t}$$

Posons $c_t(H) = \frac{f(H)}{\bar{f}_t} - 1$. On obtient alors :

$$m(H, t+1) = (1 + c_t(H))m(H, t)$$

Il est donc clair qu'un schéma, dont l'adaptation est au-dessus de la moyenne, voit son nombre de représentants augmenter, suivant une progression qui est de type

géométrique si nous faisons l'approximation que $c_t(H)$ est constant dans le temps. Alors :

$$m(H, t) = (1 + c(H))^t . m(H, 0)$$

Si la reproduction seule était en jeu, les schémas forts élimineraient très rapidement les schémas faibles.

E.1.3 Effets des croisements

Nous allons nous intéresser à la probabilité de survie $p_s(H)$ d'un schéma H lors d'une opération de croisement. Par exemple, soit le schéma $H = **10*1*$. Supposons qu'une instance de ce schéma soit croisée avec une autre instance. Quelle est la probabilité pour que la séquence résultante soit encore une instance de H ? Il est impossible de répondre exactement à la question, tout au plus peut-on donner une borne inférieure de cette valeur. Il est clair que H ne sera pas détruit si le site de croisement qui est tiré au sort est inférieur à 3 (avant le premier 1) ou s'il est supérieur à 6 (après le dernier 1)¹.

On voit donc immédiatement qu'une borne inférieure de la probabilité de détruire un schéma H est $\delta(H)/(l-1)$. Donc la probabilité de survie dans un croisement est $1 - \delta(H)/(l-1)$. Si d'autre part on ne croise qu'une fraction p_c de séquences dans une population donnée, la probabilité de survie est donnée par :

$$p_s \geq 1 - p_c \frac{\delta(H)}{l-1}$$

De ce résultat et des résultats précédents découle la loi d'évolution d'une population² :

$$m(H, t+1) \geq m(H, t)(1 + c_t(H))(1 - p_c \frac{\delta(H)}{l-1})$$

E.1.4 Effets des mutations

Soit p_m la probabilité de mutation d'un bit dans une séquence. Dans un schéma H , seules les positions fixes peuvent être détruites. La probabilité de survie d'un bit étant $1 - p_m$, la probabilité de survie d'un schéma H contenant $o(H)$ positions fixes est $(1 - p_m)^{o(H)}$. La probabilité de mutation étant supposée petite devant 1, un développement limité au premier ordre donne une probabilité de survie égale à $1 - o(H)p_m$.

L'équation finale s'écrit donc :

$$m(H, t+1) \geq m(H, t)(1 + c_t(H))(1 - p_c \frac{\delta(H)}{l-1} - o(H)p_m)$$

1. Dans tous les autres cas, H peut être (ou ne pas être) détruit.

2. Les croisements et les mutations se font sur la population reproduite, et non sur la population initiale.

E.1.5 Conclusions

Des calculs précédents découlent deux résultats :

- les schémas dont la longueur fondamentale est petite sont plus favorisés que les autres, lors de la génération d’une nouvelle population.
- les schémas dont l’ordre est petit sont plus favorisés que les autres, lors de la génération d’une nouvelle population.

Ces résultats conditionnent la qualité du codage des données. En effet, les schémas qui codent les données “intéressantes” pour le problème considéré doivent avoir un ordre et une longueur fondamentales faibles³, alors que les données “sans intérêt” doivent être codées par des schémas qui ont un ordre et une longueur fondamentale élevés.

Les résultats présentés ci-dessus ne sont qu’une introduction à la théorie des schémas, il existe de nombreux approfondissements. On trouvera dans les références suivantes les principaux modèles théoriques sur la théorie des schémas et ses extensions : [Gol89c, Vos91, BM93, Gol89b, BG87, CS88, BM94, DM92, SGE91, AS92]

E.2 Modélisation par chaîne de Markov

Cette dernière approche est la plus satisfaisante tant sur le plan mathématique, que sur celui de la modélisation, les différents opérateurs étant présentés comme “perturbant” un processus Markovien représentant la population à chaque étape. Ici encore il est démontré l’importance de l’opérateur de mutation, le croisement pouvant être totalement absent. Nous présentons ici une version simplifiée de cette théorie et les principaux résultats de convergence.

E.2.1 Modélisation de l’algorithme génétique

Les principaux résultats asymptotiques portant directement sur les algorithmes génétiques, ont été développés par R. Cerf [Cer94] sur la base des travaux de Catoni [Cat90] et de Trouvé [Tro93]. Ces travaux se fondent sur la théorie des petites perturbations aléatoire d’un processus dynamique de type Markovien. Plus particulièrement, la théorie de Freidlin et Wentzell [FW83] constitue la pierre d’angle de ces études. Nous donnons ici, quelques résultats particulièrement révélateurs de la dynamique des algorithmes génétiques, développés par Cerf. Nous les présentons simplifiés et dans un cadre restreint, laissant le lecteur intéressé se reporter à la difficile lecture des références citées ici.

Afin de préciser le cadre de cette section, nous travaillerons ici sur la base d’un codage binaire, P représentant le nombre de bits utilisés pour le codage. La fonction d’évaluation, f sera donc définie sur l’espace $E = \{0, 1\}^P$ à valeurs dans

3. Les données intéressantes sont bien entendu les données qui sont proches de la solution optimale. Un bon codage des données implique donc d’avoir une *idée* de la forme de l’optimum.

\mathcal{R}^+ . Le problème est donc de localiser l'ensemble des maxima globaux de f , ou, à défaut, de trouver rapidement et efficacement des régions de l'espace, où se situent ces maxima.

Comme nous l'avons vu l'algorithme génétique est un algorithme stochastique itératif qui opère sur des ensembles de points, et qui est bâti à l'aide de trois opérateurs : mutation, croisement et sélection, que nous présentons plus formellement à présent.

E.2.2 Description rapide de l'algorithme

Soit N la taille (fixe) de la population, notons X_k la population de la génération k : il s'agit d'une matrice $X_k = (X_k^1, X_k^2, \dots, X_k^N)$ de E^N dont les N éléments sont des chaînes de bits (chromosomes) de taille P . Le passage de la génération k à la génération $k + 1$, c'est à dire de X_k à X_{k+1} se décompose en trois étapes :

$$X_k \xrightarrow{\text{Mutation}} Y_k \xrightarrow{\text{Croisement}} Z_k \xrightarrow{\text{Sélection}} X_{k+1}$$

Chacune de ces étapes peut être modélisée formellement.

Mutation $X_k \longrightarrow Y_k$

L'opérateur considéré ici est le suivant : pour chaque composante de chaque élément X_k^i , une variable de Bernoulli de paramètre P_m est tirée indépendamment et suivant le résultat l'élément binaire examiné est changé ou non. (0 est changé en 1 et 1 en 0).

La probabilité P_m de mutation doit être préalablement choisie et est généralement faible.

Comme nous le verrons par la suite, cet opérateur joue un rôle clé dans la convergence de l'algorithme génétique.

Croisement $Y_k \longrightarrow Z_k$

L'opérateur étudié ici est l'opérateur à un point (slicing crossover). Ici encore, la probabilité de croisement P_c est fixée initialement. Pour construire la population Z_k , $N/2$ couples sont formés à partir de la population Y_k (par exemple en appariant les individus consécutifs de Y_k , ou bien en choisissant au hasard et uniformément des individus dans Y_k). Pour chaque couple, une variable de Bernoulli de paramètre P_c est tirée pour décider si le croisement a lieu. Si c'est le cas, un site de coupure est tiré au hasard, et les segments finaux des deux chromosomes sont échangés.

Une nouvelle paire d'individus est ainsi obtenue (identique à l'ancienne s'il n'y a pas eu de croisement) et est stockée dans la population Z_k . En général, le paramètre P_c est choisi grand.

Remarquons que les opérateurs de mutation et de croisement ne font pas intervenir la fonction f , ce sont des opérateurs stochastiques d'exploration. C'est le troisième et dernier opérateur, la sélection, qui guide la population vers les valeurs élevées de la fonction f .

Sélection $Z_k \longrightarrow X_{k+1}$

Les N individus de la population X_{k+1} sont obtenus après sélection des individus de Z_k . On conserve ainsi les “meilleurs” individus de Z_k , indépendamment à l’aide d’une distribution de probabilité qui favorise les individus de Z_k les mieux adaptés.

Le choix le plus fréquent est l’unique distribution telle que la probabilité d’un individu soit proportionnelle à son adaptation, i.e la probabilité de sélection de l’individu Z_k^i est :

$$P_i = P(Z_k^i) = \frac{f(Z_k^i)}{\sum_{j=1}^N f(Z_k^j)}$$

En tirant les individus dans la population Z_k conformément aux probabilités P_i , on constitue la nouvelle génération X_{k+1} .

E.2.3 Modélisation

La présentation rapide des opérateurs nous permet de modéliser la suite des $(X_k)_{k \in \mathcal{N}}$ en une chaîne de Markov, d’espace d’états $E = (\{0, 1\}^P)^N$. L’algorithme génétique ne doit donc pas être interprété comme une procédure d’optimisation mais plutôt comme une marche aléatoire dans l’espace d’états, attirée vers les fortes valeurs de f .

La propriété première de cette formalisation est que la loi de X_k est déterminée de manière unique par :

- la loi de la génération initiale X_0
- le mécanisme de transition de X_k à X_{k+1} , mécanisme scindé en trois étapes détaillées précédemment.

Ce mécanisme de transition possède toutefois des propriétés essentielles qui font l’intérêt et la puissance de cette formalisation, (voir [Cer94]) :

- Il est homogène (c’est à dire indépendant de la génération, k , considérée)
- Il est irréductible, (la probabilité de joindre deux points quelconques de l’espace d’états, en un nombre fini de générations est non nulle) soit :

$$\forall x, y \in E \quad \exists r \in \mathcal{N} \quad P[X_{k+r} = y \mid X_k = x] > 0$$

Le mécanisme permet donc d’explorer tout point de l’espace d’états, avec une probabilité non nulle.

- Il est apériodique, cette hypothèse n’est cependant pas fondamentale.

Ces propriétés permettent de conclure à l’ergodicité de la chaîne de Markov, et à l’existence d’un processus limite.

Théorème E.1 –

Une chaîne de Markov homogène irréductible apériodique d’espace d’états fini est ergodique et possède une unique mesure de probabilité stationnaire ou invariante.

Cette mesure stationnaire correspond à la loi régissant l'équilibre du processus, elle est définie, pour tout y , comme :

$$\mu(y) = \lim_{k \rightarrow \infty} P[X_k = y \mid X_0 = x]$$

Nous savons également que tout élément de l'espace d'états est de probabilité non nulle pour cette mesure.

Toutefois, si ce résultat nous permet de savoir qu'il existe une dynamique de fond de l'algorithme génétique, il nous reste à en déterminer les propriétés, l'influence des opérateurs (et des paramètres associés) qui jouent un grand rôle dans le processus.

Pour cela nous introduisons les notations suivantes :

Notations

Si $x = (x_1, \dots, x_N)$ est un élément de E^N et i un point de E , nous noterons :

$$\begin{aligned} \widehat{f}(x) &= \widehat{f}(x_1, \dots, x_N) = \max \{f(x_i) : 1 \leq i \leq N\} \\ \widehat{x} &= \{x_k \in \arg \max f(x)\} \\ [x] &= \{x_k : 1 \leq k \leq N\} \end{aligned}$$

De manière générale, les lettres $z, y, z, u, v..$ désignent des populations, i.e. des éléments de E^N , et les lettres i, j des points de E .

Processus de fond (X_k^∞)

C'est à partir de ce processus de fond qu'est reconstitué l'algorithme génétique en étudiant ses perturbations aléatoires par les différents opérateurs. Il est défini comme processus limite, lorsque les perturbations ont disparu. C'est également une chaîne de Markov sur E^N dont le mécanisme de transition est très simple puisque correspondant à la situation limite suivante :

Les N composantes de X_{k+1}^∞ sont choisies indépendamment et suivant la loi uniforme sur l'ensemble \widehat{X}_k^∞ .

- Les individus dont l'adaptation n'est pas maximale en k , sont éliminés et n'apparaissent pas dans la génération $k + 1$,
- Les individus dont l'adaptation est maximale, ont des chances de survies égales

Cette chaîne est tout d'abord piégée dans l'ensemble S des populations ayant la même adaptation (ou ensemble des population d'*équi-adaptation*),

$$S = \left\{ x = (x_1, \dots, x_N) \in E^N \quad : \quad f(x_1) = f(x_2) = \dots = f(x_N) \right\}$$

Cette population représente les *attracteurs* de la chaîne (voir E.2.4 plus loin), puis elle est absorbée par une population uniforme, de sorte que :

$$\forall x \in E^N \quad P[\exists x_i \in \widehat{x} \quad \exists K \quad \forall k \geq K \quad X_k^\infty = x_i \mid X_0^\infty = x_{ini}] = 1$$

Lorsque la population est devenue uniforme et en l'absence ici de perturbations, il ne se passe plus rien.

Ceci peut également se traduire en définissant les populations uniformes comme les *états absorbants* de la chaîne X_k^∞ .

Nous allons maintenant étudier la situation où ce processus est perturbé.

Processus perturbé (X_k^l)

La modélisation proposée par Cerf, part du processus de fond (X_k^∞), décrit ci-dessus, qui est perturbé aléatoirement, les perturbations sont indicées par le paramètre l . La chaîne de Markov (X_k^∞) devient donc une suite de chaînes de Markov (X_k^l), dont le mécanisme de transition est donné par la succession des transformations générées par les opérateurs.

$$X_k^l \xrightarrow{\text{Mutation}} U_k^l \xrightarrow{\text{Croisement}} V_k^l \xrightarrow{\text{Sélection}} X_{k+1}^l$$

Il nous faut pour cela modéliser précisément les opérateurs.

Mutation $X_k^l \rightarrow U_k^l$

Les mutations sont définies comme de petites perturbations aléatoires indépendante des individus, de la population X_k^l . Il est assez naturel d'introduire la probabilité $p_l(i, j)$ de transition⁴ de mutation entre les points i et j de E , comme un noyau Markovien p_l .

Trivialement on a :

$$\forall i \in E \quad \sum_{j \in E} p_l(i, j) = 1$$

Sur la chaîne X_k^l , la probabilité de transition entre les points x et u de E^N est :

$$P \left[U_k^l = u \mid X_k^l = x \right] = p_l(x_1, u_1) \cdot p_l(x_2, u_2) \cdots p_l(x_N, u_N)$$

Plus précisément et afin d'analyser la dynamique de (X_k^l) lorsque l tend vers l'infini, nous reportons ici les hypothèses sur le mode et la vitesse de convergence des probabilités de transition. Pour cela nous supposons l'existence d'un noyau irréductible, α , sur E , i.e. :

$\forall i, j \in E, \exists i_0, i_1, \dots, i_r$ (c'est à dire un chemin dans E) tels que $i_0 = i$ et $i_r = j$ tels que :

$$\prod_{0 \leq s \leq r-1} \alpha(i_s, i_{s+1}) > 0$$

4. C'est la probabilité $P_l(i, j)$ pour un point i de E de se transformer par mutation en un point j de E

L'hypothèse d'irréductibilité du noyau α est essentielle, elle assure que tout point de l'espace est potentiellement visitable.

La vitesse de convergence du noyau p_l , est caractérisée par le réel positif \mathbf{a} , tel que p_l admette le développement suivant :

$$\forall i, j \in E \quad \forall s \quad p_l(i, j) = \begin{cases} \alpha(i, j) \cdot l^{-\mathbf{a}} + o(l^{-s}) & \text{si } i \neq j \\ 1 - \alpha(i, j) \cdot l^{-\mathbf{a}} + o(l^{-s}) & \text{si } i = j \end{cases} \quad (\text{E.1})$$

La condition de positivité de \mathbf{a} nous permet de faire disparaître les perturbations lorsque l tend vers l'infini.

$$\forall i, j \in E \quad \lim_{l \rightarrow \infty} p_l(i, j) = \delta(i, j) = \begin{cases} 0 & \text{si } i \neq j \\ 1 & \text{si } i = j \end{cases} \quad (\text{E.2})$$

Croisement $U_k^l \rightarrow V_k^l$

Ici encore l'opérateur est modélisé comme effectuant de petites perturbations aléatoires sur des couples de la population U_k^l . Ces couples sont ici formés par les éléments successifs de la population, les transitions sont gérées par le noyau Markovien q_l sur $E \times E$, cette fois, de sorte que :

$$P \left[V_k^l = v \mid U_k^l = u \right] = q_l((u_1, u_2) \cdot (u_3, u_4) \cdots (u_{N-1}, u_N))$$

Pour ce noyau q_l nous supposons l'existence d'un noyau irréductible β sur $E \times E$, la vitesse de convergence est alors paramétrée par le réel positif \mathbf{b} tel que :

$$\forall (i_1, j_1) \in E \times E \quad \forall (i_2, j_2) \in E \times E \quad \forall s$$

$$q_l((i_1, j_1), (i_2, j_2)) = \begin{cases} \beta((i_1, j_1), (i_2, j_2)) \cdot l^{-\mathbf{b}} + o(l^{-s}) & \text{si } (i_1, j_1) \neq (i_2, j_2) \\ 1 - \beta((i_1, j_1), (i_2, j_2)) \cdot l^{-\mathbf{b}} + o(l^{-s}) & \text{si } (i_1, j_1) = (i_2, j_2) \end{cases} \quad (\text{E.3})$$

L'évanouissement asymptotique des croisements est également imposée par la positivité de \mathbf{b} :

$$\forall i_1, i_2, j_1, j_2 \in E \quad \lim_{l \rightarrow \infty} q_l((i_1, i_2)(j_1, j_2)) = \delta(i_1, i_2) \cdot \delta(j_1, j_2) \quad (\text{E.4})$$

Sélection $V_k^l \longrightarrow X_{k+1}^l$

C'est l'opérateur le plus compliqué et également le plus important puisqu'il permet la convergence vers les optima de f .

Il est modélisé à l'aide d'une fonction de sélection F_l dont Cerf nous donne une définition précise, pouvant être résumée par :

$$F_l : \{1, \dots, N\} \times (\mathbb{R}^+)^N \longrightarrow [0, 1]$$

$$(i, f_1, f_2, \dots, f_N) \longrightarrow F_l(i, f_1, f_2, \dots, f_N)$$

telle que :

- i) $F(\cdot, f_1, f_2, \dots, f_N)$ est une probabilité sur $\{1, \dots, N\}$
- ii) Cette probabilité est indépendante de l'indexation des f_1, f_2, \dots, f_N (on peut permuter les f_i)
- iii) La probabilité favorise les éléments i associés à des valeurs élevées (i.e.)

Si $f_1 \geq f_2 \geq \dots \geq f_N$ Alors

$$F_l(1, f_1, f_2, \dots, f_N) \geq F_l(2, f_1, f_2, \dots, f_N) \geq \dots \geq F_l(N, f_1, f_2, \dots, f_N)$$

Cet outil nous permet d'écrire la probabilité de transition correspondant à la dernière étape.

$$P[X_{k+1}^l = x \mid V_k^l = v] = \prod_{r=1}^N \Upsilon_l(x_r, v_r)$$

Ceci signifie que la probabilité de transition est le produit des probabilités sur chacune des N composantes de E .

La probabilité Υ_l entre deux composantes (x_r, v_r) est donnée par :

$$\Upsilon_l(x_r, v_r) = \sum_{k: v_k=x_k} F_l(k, f(v_1), f(v_2), \dots, f(v_N))$$

De même que pour les autres opérateurs, la fonction de sélection doit être choisie et sa vitesse de convergence caractérisée :

$$F_l(i, f_1, f_2, \dots, f_N) = \frac{\exp(\mathbf{c} \cdot f_i \cdot \ln(l))}{\sum_{r=1}^N \exp(\mathbf{c} \cdot f_r \cdot \ln(l))} \quad (\text{E.5})$$

Ce choix correspond bien à une probabilité de sélection avantagant les fortes adaptations au détriment des faibles, le réel positif c indexant cette fonction.

Le mécanisme de sélection opérant sur le processus de fond (X_k^∞) , correspond à la fonction de sélection F_∞ définie par :

$$F_\infty(k, f(x_1), f(x_2), \dots, f(x_N)) = \frac{\mathbf{1}_{\hat{x}}(x_k)}{\text{card}(\hat{x})}$$

C'est à dire, la loi uniforme sur l'ensemble $\hat{x} = \{x_k \in \arg \max f(x)\}$

La suite $(F_l)_{l \in \mathcal{N}}$ des fonctions de sélection tend vers cette loi uniforme

$$\forall x \in E^N \quad \forall k$$

$$\lim_{l \rightarrow \infty} F_l(k, f(x_1), f(x_2), \dots, f(x_N)) = F_\infty(k, f(x_1), f(x_2), \dots, f(x_N)) \quad (\text{E.6})$$

Les conditions E.2, E.4, et E.6 nous permettent d'assurer que le mécanisme de transition de la chaîne (X_k^l) converge vers celui du processus de fond (X_k^∞) :

$$\forall y, z \in E^N \quad \lim_{l \rightarrow \infty} P[X_{k+1}^l = z \mid X_k^l = y] = P[X_{k+1}^\infty = z \mid X_k^\infty = y]$$

C'est également en ce sens que l'on interprète la chaîne (X_k^l) comme une perturbation de la chaîne (X_k^∞) .

Les vitesses de convergence intervenant dans chacun des opérateurs jouent un rôle important. La formulation proposée en (E.1), (E.3) et (E.5), permet un ajustement équitable de ces vitesses (elles sont logarithmiquement du même ordre) de sorte qu'aucun opérateur ne domine les autres dans la dynamique. lorsque l tend vers l'infini, les conditions (E.2), (E.4), et (E.6) nous permettent d'assurer que le mécanisme de transition de la chaîne (X_k^l) converge vers celui du processus de fond (X_k^∞) , et on a :

$$\forall y, z \in E^N \quad \lim_{l \rightarrow \infty} P[X_{k+1}^l = z \mid X_k^l = y] = P[X_{k+1}^\infty = z \mid X_k^\infty = y]$$

La chaîne (X_k^l) se comporte alors comme le ferait (X_k^∞) . La théorie de Freidlin-Wentzell nous donne les outils pour simplifier l'étude de ces processus à temps continu.

E.2.4 La théorie de Freidlin et Wentzell

Principe

Soit le système différentiel de \mathfrak{R}^N satisfaisant les équations déterministes :

$$\begin{cases} dx_t = b(x_t) dt \\ x_0 = x_{ini} \end{cases} \quad (\text{E.7})$$

Sous de "bonnes" hypothèses, il existe une solution (trajectoire) unique, $x(t)$ à l'équation (E.7) et à la condition initiale associée. L'une des préoccupations immédiates est de savoir si cette solution va, ou non, tendre vers un équilibre (qui n'est pas forcément unique). Et si oui, quel en est l'ensemble de stabilité. L'équilibre est défini comme une fonction constante x^* telle que $x^* = \lim_{t \rightarrow \infty} x_t$, et l'ensemble de stabilité comme l'ensemble $K(x^*)$ des points de départ qui mènent à

cet équilibre⁵. On peut élargir cette notion, d'équilibre et de stabilité, par celles, très proches, d'attracteur et de bassin d'attraction.

Un attracteur du système est le voisinage compact K_i d'un point visité une infinité de fois, et le bassin d'attraction l'ensemble des points de départ qui mènent à cet attracteur. Nous supposons que \mathfrak{R}^d possède un nombre fini d'attracteurs K_1, \dots, K_r .

La théorie de Freidlin-Wentzell étudie l'évolution du système E.7 lorsqu'il subit des perturbations Browniennes, d'intensité ε . Le système déterministe E.7 devient alors un système différentiel stochastique.

$$\begin{cases} dX_t^\varepsilon = b(X_t^\varepsilon) dt + \varepsilon d\omega_t \\ X_0^\varepsilon = x_{ini} \end{cases} \quad (\text{E.8})$$

Le processus $(X_t^\varepsilon)_{t \in \mathfrak{R}^+}$ est maintenant un processus stochastique perturbé par le mouvement brownien $(\omega_t)_{t \in \mathfrak{R}^+}$ et dépendant de ε . La situation change alors puisque les perturbations browniennes permettent au processus de s'échapper de n'importe quel bassin d'attraction, et en fait le processus les visite tous.

De plus, le processus est ergodique et admet une unique mesure de probabilité invariante, i.e.

$$\forall \mathcal{B} \text{ borelien } \in \mathfrak{R}^d \quad \lim_{t \rightarrow \infty} P[X_t^\varepsilon \in \mathcal{B} \mid X_0^\varepsilon = x_{ini}] = \mu^\varepsilon(\mathcal{B})$$

existe et $\mu^\varepsilon(\mathcal{B})$ est la probabilité de présence du processus dans le Borélien \mathcal{B} , lorsque le système a atteint son état d'équilibre. Cette probabilité μ^ε est invariante avec le point de départ x_{ini} .

Lorsque les perturbations cessent, le processus se comporte comme dans E.7 et reste presque sûrement au voisinage $V(K_1 \cup \dots \cup K_r)$ des attracteurs, tandis que la probabilité de présence dans n'importe quel Borélien \mathcal{A} disjoint de $K_1 \cup \dots \cup K_r$ disparaît.

$$\lim_{\varepsilon \rightarrow 0} \mu^\varepsilon(V(K_1 \cup \dots \cup K_r)) = 1$$

$$\lim_{\varepsilon \rightarrow 0} \mu^\varepsilon(\mathcal{A}) = 0$$

Le résultat principal de Freidlin et Wentzell repose sur l'équivalence du processus $(X_t^\varepsilon)_{t \in \mathfrak{R}^+}$ à temps continu et espace d'états \mathfrak{R}^d et du processus $(Z_n^\varepsilon)_{n \in \mathcal{N}}$ à temps discret et espace d'états fini $\{1, \dots, r\}$ décrivant les visites au n ième attracteur.

5. L'ensemble de stabilité de l'équilibre x^* est :

$$K(x^*) = \left\{ x_{ini} \in \mathfrak{R}^d, \text{ t.q. pour } x_t \text{ solution de E.7 ; } \lim_{t \rightarrow \infty} x_t = x^* \right\}$$

Pour chaque équilibre on définit ainsi son ensemble de stabilité. Cet équilibre est stable s'il contient un voisinage de l'équilibre, et instable s'il existe des points de départ infiniment proches de l'équilibre qui ne mènent pas à celui-ci.

La construction précise de $(Z_n^\varepsilon)_{n \in \mathcal{N}}$, n'est pas détaillée ici mais nous en donnons un aperçu afin de mieux comprendre ce dernier processus.

- Si x_{ini} est “proche” de l'attracteur K_h alors $Z_0^\varepsilon = h \in \{1, \dots, r\}$
- puis le processus, sous l'influence de (ω_t) , est attiré par K_s et $Z_1^\varepsilon = s$
- etc..

La chaîne de Markov⁶ ainsi créée a pour espace d'états $\{1, \dots, r\}$, est irréductible, et possède une unique mesure de probabilité invariante ν^ε .

Théorème E.2 – *L'étude du comportement asymptotique de la mesure μ^ε est “équivalente” à l'étude du comportement asymptotique de la mesure ν^ε*

Nous passons sous silence l'étude des probabilité de transition $P[Z_n^\varepsilon = i \mid Z_n^\varepsilon = j]$ de la chaîne $(Z_n^\varepsilon)_{n \in \mathcal{N}}$ qui s'écrivent comme des intégrales sur l'ensemble des fonctions ϕ qui lient les attracteurs K_i et K_j , laissant le lecteur intéressé se reporter à la lecture de Freidlin et Wentzell, ou de Cerf.

Notons toutefois que ces probabilité de transition s'écrivent :

$$P[Z_n^\varepsilon = i \mid Z_n^\varepsilon = j] \underset{\ln}{\sim} \exp - \frac{V(i, j)}{\varepsilon^2}$$

où $V(i, j) = \inf\{V(\phi), \phi(\cdot)$ continue $[0, 1] \mapsto \mathbb{R}^d, \phi(0) \in K_i, \phi(1) \in K_j\}$

et $V(\phi) = \int_0^1 \left| \dot{\phi}(t) - \dot{b}(\phi(t)) \right|^2 dt$. est une constante associée à ϕ et caractérisant sa vitesse de convergence.

La quantité $V(i, j)$ ou *coût de communication*, mesure le coût de passage de l'attracteur K_i à l'attracteur K_j .

Les intensités de transitions de la chaîne $(Z_n^\varepsilon)_{n \in \mathcal{N}}$, nous ouvrent la voie pour déterminer la mesure invariante ν^ε .

Mesure invariante ν^ε

Les outils qui permettent de déterminer cette mesure invariante ont été développés, une nouvelle fois par Freidlin et Wentzell, nous aurons besoin de certains d'entre eux.

Définition : Soit i un élément de $\{1, \dots, r\}$.

Un i -*graphe* sur $\{1, \dots, r\}$ est un graphe g sur $\{1, \dots, r\}$ possédant les propriétés suivantes :

- $\forall j \neq i$, le graphe g contient une unique flèche issue de j
- Il existe un chemin dans g qui mène de j à i
- g ne contient pas de flèche issue de i

6. La nature Markovienne de ω_t , nous permet de montrer qu'il s'agit bien là d'une chaîne de Markov.

Il s'agit donc d'un graphe sans cycles formé de chemins qui aboutissent en i . On note $G(i)$ l'ensemble des i -graphes sur $\{1, \dots, r\}$.

Définition : La fonction d'énergie virtuelle W est la fonction de $\{1, \dots, r\}$ dans \mathbb{R}^+ définie par :

$$\forall i \in \{1, \dots, r\} \quad W(i) = \min_{g \in G(i)} \sum_{(\alpha \rightarrow \beta) \in g} V(\alpha, \beta)$$

A cette fonction est associé l'ensemble W^* des minima globaux de W .

Finalement, la mesure invariante ν^ε est caractérisée par :

$$\forall i \in \{1, \dots, r\} \quad \nu^\varepsilon(i) \underset{\ln}{\sim} \exp - \frac{W(i) - W(W^*)}{\varepsilon^2}$$

où $W(W^*) = \min \{W(i) : i \in \{1, \dots, r\}\}$.

Le comportement asymptotique de ν^ε (et par la même occasion de μ^ε) est donc connu : la mesure ν^ε se concentre sur les attracteurs dont l'indice est dans W^* et décroît vers zéro à la vitesse $\exp - \frac{Cste}{\varepsilon^2}$ pour les autres attracteurs. Il existe donc un sous-ensemble de W^* de l'ensemble des attracteurs sur lequel se concentre la mesure invariante du processus.

$$\lim_{\varepsilon \rightarrow 0} \lim_{t \rightarrow \infty} P \left[X_t^\varepsilon \in V \left(\bigcup_{i \in W^*} K_i \right) \mid X_0^\varepsilon = x_{ini} \right] = 1$$

La dynamique du processus est donc caractérisable.

Dynamique du processus.

Dans sa thèse, Cerf nous donne une très claire interprétation de la hiérarchie des cycles qui caractérisent la dynamique du processus. Supposons que le processus soit initialement dans le bassin d'attraction de K_1 . Il quitte K_1 au bout d'un temps fini. Parmi toutes les trajectoires de sortie, il en existe une plus "probable" que les autres, qui l'amène vers un nouvel attracteur ; par exemple K_2 puis, bientôt K_3 . L'ensemble des attracteurs étant par hypothèse fini, le processus finit par revisiter un attracteur formant un cycle d'ordre 1 sur lequel le processus tourne longtemps, très longtemps. Englobons maintenant ces trois attracteurs dans une boîte. Comme toujours, les perturbations browniennes finissent par pousser le processus hors de cette boîte, et ici encore, il existe une trajectoire de sortie canonique qui fait tomber le processus dans un nouveau bassin d'attraction, ou plus généralement, dans un autre cycle d'ordre 1.

Les cycles d'ordre 1 sont aussi en nombre fini, et le processus finit par revisiter un cycle d'ordre 1 : un cycle d'ordre 2 est alors formé, dans lequel le processus reste piégé très longtemps. En continuant de la sorte, il est possible de construire

toute une hiérarchie de cycles qui épuise l'ensemble des attracteurs et fournit une image très précise de la dynamique asymptotique du processus. A chaque transition entre cycles est associée une constante qui caractérise la difficulté de la transition.

Enfin, lorsque ε décroît avec le temps ($\varepsilon = \varepsilon(t)$ est une fonction de t qui tend en décroissant vers 0), nous obtenons un processus de Markov inhomogène (le mécanisme de transition dépend du temps).

- Si $\varepsilon(t)$ décroît très lentement, de sorte qu'à chaque instant la loi de X_t soit proche de l'état d'équilibre associé au niveau de perturbation $\varepsilon(t)$, la situation ne change pas fondamentalement. La loi limite correspond à la limite de la suite des lois d'équilibre.
- Si au contraire $\varepsilon(t)$ décroît très rapidement, le processus risque de rester piégé dans certains sous-ensembles d'attracteurs : plus précisément, dans la hiérarchie des cycles, certaines transitions ne pourront être effectuées qu'un nombre fini de fois, alors que d'autres, plus "faciles", seront réalisées une infinité de fois avec probabilité 1. La loi limite dépend alors fortement du point de départ.

La hiérarchie des cycles permet ainsi de décrire les dynamiques possibles de (X_t) en fonction de la vitesse de décroissance de $\varepsilon(t)$.

E.2.5 Résultats de convergence

Lorsque l croit vers l'infini, les perturbations affectant le processus (X_k^l) diminuent de sorte que cette chaîne se comporte, presque sûrement, comme la chaîne (X_k^∞) . Plus précisément, nous savons que les attracteurs de la chaîne (X_k^∞) sont les populations d'équi-adaptation S et les populations uniformes (attracteurs stables). La chaîne (X_k^l) va donc être attirée par ses attracteurs, en commençant par l'ensemble S .

La théorie de Freidlin et Wentzell nous permet de reporter cette étude sur celle de la chaîne des (Z_k^l) des visites successives de (X_k^l) à l'ensemble S . Nous posons donc $Z_k^l = X_{T_k}^l$ où T_k est l'instant de la k ième visite de (X_k^l) dans S .

Les probabilités de transition de la chaîne (Z_k^l) , sont estimées à l'aide des opérateurs définis en E.2.3 et selon le schéma développé ci-dessus. Les fonctions de coût de communication $V(i, j)$ et d'énergie virtuelle W sont définies et estimées.

Nous savons alors que la suite des mesures stationnaires de la chaîne (X_k^l) se concentre sur l'ensemble W^* des minima de W :

$$\forall x_{ini} \in E^N \quad \lim_{l \rightarrow \infty} \lim_{k \rightarrow \infty} P \left[X_k^l \in W^* \mid X_0^l = x_{ini} \right] = 1$$

L'un des principaux résultats indique qu'il existe une taille de la population de (X_k^l) , (taille critique) telle que les maxima de f soient atteints asymptotiquement avec la probabilité 1.

Taille critique

Supposons fixés l'espace d'états E , la fonction d'adaptation f , les noyaux de transition de mutation α et de croisement β , ainsi que les constantes positives gé-rant les trois opérateurs \mathbf{a} , \mathbf{b} , et \mathbf{c} .

Théorème E.3 – (Cerf 1993)

Il existe une valeur critique N^ , telle que lorsque la taille de la population de l'algorithme génétique dépasse N^* , l'ensemble f^* des maxima globaux de f , contient l'ensemble W^* .*

Cette taille critique N^ , dépend fortement de l'espace d'états E , de la fonction d'adaptation f , des noyaux de transition de mutation α et de croisement β , ainsi que des paramètres \mathbf{a} , \mathbf{b} , et \mathbf{c} .*

Une borne grossière, mais lisible de N^* est :

$$N^* \leq \frac{\mathbf{a}R + \mathbf{c}(R - 1)\Delta}{\min(\mathbf{a}, \frac{\mathbf{b}}{2}, \mathbf{c}\delta)}$$

où :

- R est le nombre minimal de transition permettant de joindre deux points arbitraires de E par mutation
- Δ et δ sont des paramètres d'échelle :
 - $\Delta = \max \{|f(i) - f(j)| : i, j \in E\}$ paramètre mesurant les écarts maximaux de f
 - $\delta = \min \{|f(i) - f(j)| : i, j \in E, f(i) \neq f(j)\}$ mesurant les écarts minimaux de f .

Il est intéressant de relever que le résultat est obtenu sans faire intervenir l'opérateur de croisement, qui n'est donc pas indispensable. L'exploration par mutation et la sélection suffisent à assurer la convergence vers f^* ([Zhi91]).

Ce premier résultat nous indique que dès que $N \geq N^*$, la suite des mesures stationnaires de la chaîne (X_k^l) se concentre asymptotiquement sur f^* lorsque l tend vers l'infini. On peut dans une étape suivante faire évoluer l , et donc l'intensité des perturbations, en fonction de la génération. Nous obtenons alors une chaîne de Markov inhomogène $(X_k^{l(k)})$ dont le mécanisme de transition dépend alors de la génération k .

Vitesse de convergence

Le principal problème est de savoir si cette chaîne inhomogène peut avoir un comportement proche de celui de la chaîne homogène (X_k^l) , et si oui, sous quelles conditions. La vitesse de croissance de $l(k)$ vers l'infini, est bien évidemment, au centre du débat.

- Si $l(k)$ croît "lentement", alors la loi de X_k sera proche de la loi stationnaire $\mu^{\varepsilon(l(n))}$ de niveau de perturbation $\varepsilon(l(n))$ associé à $l(n)$.
- Si $l(k)$ croît "rapidement", alors X_k risque de rester piégé dans des bassins d'attraction ne correspondant pas aux maxima de f , l'intensité des perturbations devenant trop faible pour pouvoir s'en échapper.

La vitesse recherchée se situe entre ces deux extrêmes, permettant à X_k de s'échapper des "mauvais" bassins d'attraction (ne correspondant pas à des maxima de f) et de rester piégé dans le "bon" (celui des points de f^*).

La vitesse de convergence de la suite $l(k)$ est caractérisée par l'exposant de convergence⁷, λ .

Définition :

L'exposant de convergence λ de la suite $l(k)$ est l'unique réel λ tel que :

$$\begin{aligned} \sum_{k \in \mathcal{N}} l(k)^{-\theta} &\rightarrow \text{converge pour } \theta > \lambda \\ &\rightarrow \text{diverge pour } \theta < \lambda \end{aligned}$$

Deux conditions pour la colonisation de f^* sont également données par Cerf, l'une nécessaire, l'autre suffisante.

Théorème E.4 – Condition nécessaire pour la colonisation de f^*

Pour que :

$$\forall x_{ini} \in E^N \quad P[\exists K \quad \forall k \geq K \quad [X_{T_k}] \subset f^* \mid X_0 = x_{ini}] = 1$$

c'est à dire, pour que la chaîne $Z_k = X_{T_k}$ des visites successives des attracteurs soit piégée dans f^ après un nombre fini K de transitions, il est nécessaire que l'exposant de convergence λ de la suite $l(k)$ appartienne à l'intervalle $]\phi, \psi[$.*

Les constantes ϕ et ψ sont des caractéristiques du problème, l'intervalle $]\phi, \psi[$ est alors non vide pour N assez grand.

Théorème E.5 – Condition suffisante pour la colonisation de f^*

Il existe deux constantes η et ρ telles que si l'exposant de convergence λ de la suite $l(k)$ appartient à l'intervalle $]\eta, \rho[$, alors :

$$\forall x_{ini} \in E^N \quad P[\exists K \quad \forall k \geq K \quad [X_{T_k}] \subset f^*, \widehat{X}_k \subset f^* \mid X_0 = x_{ini}] = 1$$

ce qui signifie qu'après un nombre fini de transitions, nous avons presque sûrement, la situation suivante :

- la chaîne X_{T_k} est piégée dans f^* ,
- la population X_k contient toujours un ou des individus appartenant à f^* .

En guise de conclusion

D'autres résultats existent, tant dans le travail de Cerf, que dans la littérature citée dans cette section. Ils demandent cependant un investissement supplémentaire dans la compréhension des outils développés. Le but de cette section était de convaincre le lecteur que l'étude théorique des algorithmes génétiques donne

7. Egalement appelé *rayon de convergence*.

(déjà) de substantiels résultats. De nombreuses interrogations demeurent cependant concernant les relations réelles entre les différents paramètres caractérisant l'algorithme génétique et les choix pratiques de ceux-ci. Dans ce domaine, la pratique devance encore la théorie, même si les mécanismes commencent à être plus clairs. Il reste également à étudier les nombreux raffinements (tels que le scaling, sharing, le clustering, l'élitisme) parfois indispensables dans la pratique.

Bibliographie

- [AL92] Luc Angerand and Hervé LeJeannic. Bilan du projet SAINTEX. Technical report, CENA, 1992. CENA/R92009.
- [AS92] J.M Alliot and T Schiex. *Intelligence Artificielle et Informatique Théorique*. Cepadues, 1992. ISBN : 2-85428-324-4.
- [AS96] C. Garcia Avello and Sip Swierstra. Towards optimum air traffic management. Technical Report DED-1 DED-2, DOC. 96.70.09, EUROCONTROL, 96, Rue de la Fusée, B - 1130 Brussels, March 1996.
- [BG87] C.L Bridges and D.E Goldberg. An analysis of reproduction and crossover in a binary-coded genetic algorithm. In *Proceedings of the Second International Conference on Genetic Algorithm*. ICGA, 1987.
- [BG91] C.L Bridges and D.E Goldberg. An analysis of multipoint crossover. In *Proceedings of the Foundation Of Genetic Algorithms*. FOGA, 1991.
- [BH75] Bryson and Ho. *Applied Optimal Control*. Hemisphere Publishing Corporation, New York, 1975.
- [BH91] John T. Betts and William P. Huffman. Trajectory optimization on a parallel processor. *Journal of Guidance, Control and Dynamics*, 14(2) :431–439, 1991.
- [BM93] T.N Bui and B.R Moon. Hyperplane synthesis for genetic algorithms. In *Proceedings of the Fifth International Conference on Genetic Algorithm*. ICGA, 1993.
- [BM94] T.N Bui and B.R Moon. Analysing hyperplane synthesis in genetic algorithms using clustered schemata. Technical Report Cse-94-026, Dep. of Comp. Sci. and Engineering, Penn. State University, March 1994.
- [Cat90] O. Catoni. *Large deviations for Annealing*. PhD thesis, Université de Paris XI, 1990.
- [Cel90] Joseph C. Celio. Controller perspective of AERA2. Technical report, MITRE, February 1990. MP-88W00015.
- [Cer94] R Cerf. *Une Théorie Asymptotique des Algorithmes Génétiques*. PhD thesis, Université Montpellier II (France), 1994.

- [CGT92a] A. R. Conn, N. I. M Gould, and P. L. Toint. LANCELOT : a fortran package for large-scale nonlinear optimization (release a). *Springer Series in Computational Mathematics*, 1992.
- [CGT92b] A.R. Conn, Nick Gould, and Ph. L. Toint. A comprehensive description of LANCELOT. Technical report, IBM T.J. Watson research center, 1992. Report 91/10.
- [Cle90] John C. Clements. Minimum-time turn trajectories to fly-to points. *Optimal Control Applications and Methods*, 11 :39–50, 1990.
- [CMMR87] A. Corana, M. Marchesi, C. Martini, and S. Ridella. Minimizing multimodal unctions of continuous variables with the “simulated annealing” algorithm. In *Proceedings of the ACM Transaction and Mathematical Software*. ACM, 1987.
- [CS88] R.A Caruana and J.D Schaffer. Representation and hidden bias : Gray versus binary coding for genetic algorithms. In *Proceedings of the Fifth International Conference on Machine Learning*, 1988.
- [DA93] Nicolas Durand and Jean-Marc Alliot. Existing algorithms for collision avoidance, and what the future might hold. Technical report, CENA, 1993.
- [DAAS94] Nicolas Durand, Nicolas Alech, Jean-Marc Alliot, and Marc Schoenauer. Genetic algorithms for optimal air traffic conflict resolution. In *Proceedings of the Second Singapore Conference on Intelligent Systems*. SPICIS, 1994.
- [DAN94] Nicolas Durand, Jean-Marc Alliot, and Joseph Noailles. Algorithmes genetiques : un croisement pour les problemes partiellement separables. In *Proceedings of the Journees Evolution Artificielle Franco-phones*. EAF, 1994.
- [DAN96a] N. Durand, J.M. Alliot, and J. Noailles. Collision avoidance using neural networks learned by genetic algorithms. In *Ninth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Fukuoka, 1996.
- [DAN96b] Nicolas Durand, Jean-Marc Alliot, and Joseph Noailles. Automatic aircraft conflict resolution using genetic algorithms. In *Proceedings of the Symposium on Applied Computing, Philadelphia*. ACM, 1996.
- [DCA96] N. Durand, O. Chansou, and J.M. Alliot. An optimizing conflict solver for atc. *ATC Quarterly*, 1996.
- [DDAS96] D. Delahaye, N. Durand, J.M. Alliot, and M. Schœnauer. Genetic algorithms for air traffic control systems. In *INFAUTOM, Toulouse*, 1996.
- [DM92] D Dasgupta and D.R McGregor. A structured genetic algorithm. Technical Report IKBS-8-92, Dep. of Computer Science. University of Strathclyde, Glasgow. UK, 1992.

- [E.K82] E.Kreindler. Additional necessary conditions for optimal control with state-variable inequality constraints. *Journal of Optimization theory and applications*, 38(2) :241–250, october 1982.
- [FMT93] Xavier Fron, Bernard Maudry, and Jean-Claude Tumelin. Arc 2000 : Automatic radar control. Technical report, Eurocontrol, 1993.
- [FOW66] L.J Fogel, A.J Owens, and M.J Walsh. *Artificial Intelligence Through Simulated Evolution*. Wiley and sons. NY, 1966.
- [FW83] M.I Freidlin and A.D Wentzell. *Random Perturbations of Dynamical Systems*. Springer-verlag, New-York, 1983.
- [GM79] M. Gondran and M. Minoux. *Graphes et Algorithmes*. Eyrolles, 1979. ISBN :0399-4198.
- [Gol89a] David Goldberg. *Genetic Algorithms*. Addison Wesley, 1989. ISBN : 0-201-15767-5.
- [Gol89b] D.E Goldberg. Genetic algorithms and walsh functions. part 1 and 2. *Complex Systems*, 3 :129–171, 1989.
- [Gol89c] D.E Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading MA Addison Wesley, 1989.
- [Gol91] D.E Goldberg. Real-coded genetic algorithms, virtual alphabets and blocking. *Complex Systems*, 5 :139–167, 1991.
- [GT82] A. Griewank and Ph. L. Toint. On the unconstrained optimization of partially separable functions. In M. J. D. Powell, editor, *Nonlinear Optimization 1981*, pages 301–312, London and New York, 1982. Academic Press.
- [Han92] E. Hansen. *Global optimization using interval analysis*. Dekker, New-York, 1992.
- [HN93] J. Horn and N. Nafpliotis. Multiobjective optimization using the niched pareto genetic algorithm. Illigal Report 93005, University of Illinois at Urbana, 1993.
- [Hol75] J.H Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan press, 1975.
- [HP95] R. Horst and P. Pardalos. *Global optimization using interval analysis*. Kluwer, Dordrecht/Boston/London, 1995.
- [HSV95] R. F. Hartl, S. P. Sethi, and R. G. Vickson. A survey of the maximum principles for optimal control problems with state constraints. *SIAM Review*, 1995.
- [Ing89] Lester Ingber. Very fast simulated re-annealing. *Mathematical and Computer Modelling*, 12 :967–973, 1989.
- [Iof84] A. D. Ioffé. Necessary conditions in nonsmooth optimization. *Mathematical Operational Research*, 1984.

- [IR92a] L Ingber and B Rosen. Genetic algorithms and very fast simulated reannealing. *Mathematical Computer Modeling*, 16(11) :87–100, 1992.
- [IR92b] Lester Ingber and Bruce Rosen. Genetic algorithm and very fast simulated reannealing : a comparison. *Mathematical and Computer Modeling*, 16(1) :87–100, 1992.
- [JKP72] S.L.S. Jacoby, J.S. Kowalik, and J.T. Pizzo. *Iterative methods for non linear optimization problems*. Prentice Hall, 1972.
- [JLS71] D. H. Jacobson, M. M. Lele, and J. L. Speyer. New necessary conditions of optimality for control problems with state-variable inequality constraints. *Journal of Mathematical, Analysis and Applications*, 35 :255–284, 1971.
- [JWS63] A.E.Bryson Jr, W.F.Denham, and S.E.Dreyfus. Optimal programming problems with inequality constraints : Necessary conditions for extremal solutions. *AIAA Journal*, 1 :2544–2550, november 1963.
- [K⁺89] Fred Krella et al. Arc 2000 scenario (version 4.3). Technical report, Eurocontrol, April 1989.
- [M⁹⁴] Frédéric Médioni. Algorithmes génétiques et programmation linéaire appliqués a la résolution de conflits aériens. Master’s thesis, Ecole Nationale de l’Aviation Civile (ENAC), 1994.
- [Mau79] H. Maurer. On the minimum principle for optimal control problems with state constraints. *Schriftenreihe des Rechenzentrums der Univ. Munster*, 1979.
- [MDA94] F. Medioni, Nicolas Durand, and J.M. Alliot. Algorithmes génétiques et programmation linéaire appliqués à la résolution de conflits aériens. In *Proceedings of the Journees Evolution Artificielle Francophones*. EAF, 1994.
- [MDA96] F. Medioni, N. Durand, and J.M. Alliot. Air traffic conflict resolution by genetic algorithms. In *Symposium on Combinatorial Optimization, CO96, London*, 1996.
- [ME88] Magdi S. Mahmoud and Shawki Z. Eid. Optimization of freeway traffic control problems. *Optimal Control Applications and Methods*, 9 :37–49, 1988.
- [MG92] S.W Mahfoud and D.E Goldberg. Parallel recombinative simulated annealing : A genetic algorithm. Illigal report 92002, University of Illinois, Urbana, IL 61801-2996, April 1992.
- [MG94] Dr. C. Meckiff and Dr. P. Gibbs. PHARE : Highly interactive problem solver. Technical report, Eurocontrol, 1994.
- [Mic92] Z Michalewicz. *Genetic algorithms + Data Structures = Evolution Programs*. Springer-verlag, 1992.

- [MJ91] Z Michalewicz and C.Z Janikov. Handling constraints in genetic algorithms. In *Proceedings of the Fourth International Conference on Genetic Algorithm*. ICGA, 1991.
- [MZ88] L. Mikhailov and J. Zaprianov. Multilevel hierarchical optimization of aircraft trajectories. In *IFAC/IMACS Symposium*, July 1988.
- [NFC⁺83] W.P. Niedringhaus, I. Frolow, J.C. Corbin, A.H. Gisch, N.J. Taber, and F.H. Leiber. Automated En Route Air Traffic Control Algorithmic Specifications : Flight Plan Conflict Probe. Technical report, FAA, 1983. DOT/FAA/ES-83/6.
- [Nie89a] W.P. Niedringhaus. Automated planning function for AERA3 : Manoeuver Option Manager. Technical report, FAA, 1989. DOT/FAA/DS-89/21.
- [Nie89b] W.P. Niedringhaus. A mathematical formulation for planning automated aircraft separation for AERA3. Technical report, FAA, 1989. DOT/FAA/DS-89/20.
- [NM65] J.A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7 :308–313, 1965.
- [PA91] Tekla S. Perry and John A. Adam. Improving the world's largest, most advanced system ! *IEEE Spectrum*, February 1991.
- [RR95] Helmut Ratschek and Jon Rokne. Interval methods. In Reiner Horst and Panos Pardalos, editors, *Handbook of global Optimization*. Kluwer, 1995.
- [RSS95] C. Ravise, Michele Sebag, and Marc Schœnauer. Optimisation guidée par l'induction. In *Proceedings of the Journées Evolution Artificielle Francophones*. EAF, 1995.
- [Sch90] Robert L. Schultz. Three-dimensional trajectory optimization for aircraft. *Journal of Guidance, Control and Dynamics*, 13(6) :936–943, 1990.
- [SGE91] R.E Smith, D.E Goldberg, and J.A Earickson. *SGA-C : A C-language implementation of a Simple Genetic Algorithm*, May 1991. TCGA report No. 91002.
- [SPSS83] E. M. Schuster, F. R. Petroski, R. K. Sciambi, and M. MC Stokrp. AERA 2 functional design and performance description. Technical report, MITRE, September 1983. MtR-83W136.
- [SSR94] Michele Sebag, Marc Schœnauer, and Caroline Ravisé. Evolution darwinienne ou évolution civilisée. In *Proceedings of the Journées Evolution Artificielle Francophones*. EAF, 1994.
- [Tro93] A. Trouvé. *Parallélisation massive du recuit simulé*. PhD thesis, Université de Paris XI, 1993.

- [vLA87] P.J.M. van Laarhoven and E.H.L. Aarts. *Simulated annealing : Theory and Applications*. Kluwer Academic Publishers, 1987. ISBN : 90-277-2513-6.
- [Vos91] M.D Vose. Generalizing the notion of schema in genetic algorithms. *Artificial Intelligence*, 50 :385–396, 1991.
- [Wri91] A.H Wright. Genetic algorithms for real parameter optimization. In *Proceeding of the Foundation Of Genetic Algorithms*. FOGA, 1991.
- [YG93] X Yin and N Gernay. A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization. In *Proceedings of the Artificial Neural Nets and Genetic Algorithms Conference*, 1993.
- [Zeg93] Karim Zeghal. Champs de forces symétriques : La logique d’un système anticollision coordonné. Technical report, ONERA, 1993.
- [Zeg94] Karim Zeghal. *Vers une théorie de la coordination d’actions, application à la navigation aérienne*. PhD thesis, Université Paris VI, 1994.
- [Zhi91] Anatoly A. Zhigljavsky. *Theory of Global Random Search*. Kluwer Academic Publishers, 1991.