UNIVERSITÉ DE MONTRÉAL

PLANIFICATION DE TRAJECTOIRES POUR UNE FLOTTE D'UAVS

ABDESSAMAD AIT EL CADI

DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION

DU DIPLÔME DE PHILOSOPHIAE DOCTOR (Ph.D.)

(GÉNIE INDUSTRIEL)

AOÛT 2010

© Abdessamad Ait El Cadi, 2010.

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

PLANIFICATION DE TRAJECTOIRES POUR UNE FLOTTE D'UAVS

présentée par : AIT EL CADI Abdessamad

en vue de l'obtention du diplôme de : Philosophiae Doctor

a été dûment accepté par le jury d'examen constitué de :

M. SOUMIS François, Ph.D., président

M. HERTZ Alain, Ph.D., membre et directeur de recherche

M. CRAINIC Teodor Gabriel, Ph.D., membre et codirecteur de recherche

M. GENDREAU Michel, Ph.D., membre

Mme ABI-ZEID Irène, Ph.D., membre externe

M. FRIGON Jean-François, Ph.D., représentant du directeur des études supérieures

DÉDICACE

À ma famille, la famille AIT EL CADI
qui était toujours là pour moi;

À mon épouse;

À la mémoire de Dr. Sophie Lapierre;

À Fattouma, Hassan, Zakia, Fatima, Souad,
Abdelfattah, Khaled, Rita et Ahmed-Réda.

الحمد لله

REMERCIEMENTS

Tous mes remerciements à mes directeurs, les professeurs Alain HERTZ et Teodor Gabriel CRAINIC. Un grand merci pour votre soutien, votre encadrement, votre présence et surtout pour votre patience malgré mes débordements. J'en profite aussi pour vos présenter toutes mes excuses pour tous les désagréments que j'aurai pu vous causer.

Un grand merci à la Défense Canadienne en la personne de M. Camille Alain RABBATH pour leur aide et leur collaboration.

Un merci spécial pour M. François SOUMIS et pour tous les membres du jury qui m'ont fait l'honneur d'accepter d'examiner ce modeste projet.

Un énorme merci à l'École Polytechnique de Montréal et aux centres GERAD et CIRRELT pour m'avoir permis de réaliser ce projet dans les meilleures conditions.

Finalement, un grand merci à toute la famille et les amis pour leur soutien inconditionnel. Tout spécialement, un merci pour Mme Zakia, SAMIR et M. Jimmy, SKELLING.

RÉSUMÉ

Les drones, qu'on appelle aussi UAVs (Unmanned Aerial Vehicles) ou RPV (Remotely Piloted Vehicles), sont des avions sans pilote. Ils sont adoptés par des organisations militaires et civiles pour accomplir des tâches difficiles dans des environnements très hostiles, sans aucun risque pour l'humain. Comme toute nation développée, le Canada a recours à ces avions pour ses missions militaires de maintien de la paix et pour des missions civiles telles que la surveillance du littoral, la surveillance des grands territoires et pour l'aide lors d'opérations de sauvetage.

L'utilisation des UAVs est très diverse, que ce soit pour des opérations militaires ou civiles, et l'engouement pour ces engins est de plus en plus grand dans les pays industrialisés. Les progrès réalisés dans les technologies de contrôle, de détection et de calcul ont permis à ces véhicules de réaliser des missions indépendantes du contrôle direct de l'opérateur. L'itinéraire peut être planifié à l'avance de la mission et le drone peut alors l'exécuter automatiquement. La planification de cet itinéraire efficace implique la détermination de solutions permettant d'atteindre un certain but fixé, comme par exemple éviter que le drone soit détecté par les radars au cours de son itinéraire, ou trouver l'itinéraire le plus court en terme de temps de déplacement, ou encore la minimisation du coût de la mission.

Dans cette thèse nous nous intéressons au problème de la planification de trajectoires pour une flotte d'UAVs lors d'une mission de surveillance. La problématique est très

vaste et touche à plusieurs disciplines. Nous avons pu couvrir trois parties importantes de cette problématique :

- la modélisation du terrain avec toutes ses contraintes;
- le calcul d'un plus court chemin continu non holonome dans un environnement risqué et en présence d'obstacles;
- la planification d'une mission de surveillance d'une flotte de drones dans un contexte réel.

Lors de l'étude de la littérature liée à ces problématiques, nous avons détecté des manques importants dans la modélisation du terrain et dans le calcul du plus court chemin continu. Ces deux manques représentent des aspects cruciaux pour l'accélération du temps de calcul lors de la planification d'une mission. Dans le cadre de cette thèse, nous avons pu apporter une contribution dans ce domaine et avons ensuite appliqué nos algorithmes à la planification d'une mission d'une flotte d'UAVs dans un environnement à risque et en présence d'obstacles. Les obstacles peuvent être naturels tels que des forêts et des montagnes ou toute zone non inaccessible pour l'avion.

Nous avons commencé par modéliser le terrain sous la forme d'un graphe orienté. Cependant, au lieu de baser notre graphe sur un maillage classique, nous avons opté pour une modélisation intelligente qui permet de réduire le temps de calcul sur ce graphe sans perdre en précision (ce qui est le problème classique des maillages). La modélisation est basée sur la notion de graphe de visibilité et intègre la prise en compte des obstacles, des zones de danger et la contrainte de non-holonomie – la contrainte de

cinématique des avions qui implique un angle maximum de braquage. Le graphe ainsi obtenu est ensuite assaini pour ne garder que le strict nécessaire pour le calcul des trajectoires. La génération du graphe peut demander beaucoup de temps de calcul, mais cette génération n'est faite qu'une seule fois avant la planification et n'affecte donc en rien les temps des calculs de trajectoires. Nous avons aussi développé un autre graphe contenant moins d'informations et ne tenant pas compte de la contrainte de non-holonomie. L'avantage de ce dernier est que le temps de sa génération est très court. Par contre, l'utilisation nécessite la mise en place d'une procédure de correction de la trajectoire pour la rendre non holonome. Cette correction est réalisable dans le contexte de nos missions mais pas pour tous les types de véhicules autonomes.

Une fois ce graphe généré, nous avons proposé une procédure de calcul d'un plus court chemin continu non holonome dans un environnement à risque et en présence d'obstacles. Étant donné que le graphe utilisé intègre déjà toutes les contraintes, nous avons modélisé le problème comme une recherche d'un plus court chemin avec une contrainte de ressource (la ressource est la quantité de risque allouée). Les résultats sont très satisfaisants puisque les itinéraires sont des trajectoires non holonomes qui respectent toutes les contraintes. De plus, le temps de calcul est très court. Pour les cas du graphe simplifié, nous avons créé une procédure de correction de la trajectoire pour la rendre non holonome. Tous les calculs de non-holonomie sont basés sur les courbes de Dubins (1957).

Nous avons finalement appliqué nos résultats à la problématique de la planification d'une mission pour une flotte d'UAVs dans un environnement à risque et en présence

d'obstacles. Nous avons pour cela développé un multi-graphe orienté qui, entre chaque paire de cibles (points de départ et de retour de la mission inclus), calcule une série de plus courtes trajectoires avec différentes nuances de risque – de la trajectoire sans risque à la trajectoire la plus risquée. Ensuite une recherche Tabou avec une double liste nous permet de trouver les routes pour chaque UAV avec pour objectif de minimiser le coût de la mission tout en respectant la limite de risque et en évitant les obstacles. Les tests sont faits sur des terrains créés à partir de descriptions données par la défense canadienne et sur certaines instances de CVRP (Capacitated Vehicle Routing Problem), celles décrites par Christofides et Elion et celles décrites par Christofides, Mingozzi et Toth. Les résultats obtenus sont très satisfaisants puisque les trajectoires respectent la non-holonomie des avions et l'amélioration de l'objectif, le coût de la mission, par rapport à un simple algorithme constructif atteint dans certains cas entre 10 % et 43 %. Nous avons même pu réaliser une amélioration de 69 % mais sur une solution initiale générée par un algorithme glouton.

ABSTRACT

UAVs (Unmanned Aerial Vehicles) or RPV (Remotely Piloted Vehicles), are planes without pilots. They were adopted by military and civilian organizations to accomplish difficult tasks over hostile environments and without risk for humans. Like any other developed nation, Canada has used the UAVs for its military missions such as peacekeeping and for civilian missions such as coastal monitoring, surveillance of large areas and rescue operations.

The applications of UAVs are very diverse, ranging from surveillance operations to combat operations through rescues. Advances in monitoring technologies, detection and computing have allowed these vehicles to perform missions without the direct control of the operator. The itinerary can be planned in advance and the UAVs can then run automatically. Planning efficient routes involves determining a specific solution with a precise objective like avoiding radar detection of the UAVs, finding the shortest path in terms of time, or minimizing the cost of the mission.

In this thesis we address the problem of coordinating and controlling a fleet of UAVs during a surveillance mission in a dynamic context. The problem is vast and is related to several scientific domains. We have studied three important parts of this problem:

- modeling the ground with all its constraints;
- computing a shortest non-holonomic continuous path in a risky environment with a presence of obstacles;

- planning a surveillance mission for a fleet of UAVs in a real context.

While investigating the scientific literature related to these topics, we have detected deficiencies in the modeling of the ground and in the computation of the shortest continuous path, two critical aspects for the planning of a mission. So after the literature review, we have proposed answers to these two aspects and have applied our developments to the planning of a mission of a fleet of UAVs in a risky environment with the presence of obstacles. Obstacles could be natural like mountain or any non flyable zone.

We have first modeled the ground as a directed graph. However, instead of using a classic mesh, we opted for an intelligent modeling that reduces the computing time on the graph without losing accuracy. The proposed model is based on the concept of visibility graph, and it also takes into account the obstacles, the danger areas and the constraint of non-holonomy of the UAVs- the kinematic constraint of the planes that imposes a maximum steering angle. The graph is then cleaned to keep only the minimum information needed for the calculation of trajectories. The generation of this graph possibly requires a lot of computation time, but it is done only once before the planning and will not affect the performance of trajectory calculations. We have also developed another simpler graph that does not take into account the constraint of non-holonomy. The advantage of this second graph is that it reduces the computation time. However, it requires the use of a correction procedure to make the resulting trajectory

non-holonomic. This correction is possible within the context of our missions, but not for all types of autonomous vehicles.

Once the directed graph is generated, we propose the use of a procedure for calculating the shortest continuous non-holonomic path in a risky environment with the presence of obstacles. The directed graph already incorporates all the constraints, which makes it possible to model the problem as a shortest path problem with resource a resource constraint (the resource here is the amount of permitted risk). The results are very satisfactory since the resulting routes are non-holonomic paths that meet all constraints. Moreover, the computing time is very short. For cases based on the simpler graph, we have created a procedure for correcting the trajectory to make it non-holonomic. All calculations of non-holonomy are based on Dubins curves (1957).

We have finally applied our results to the planning of a mission of a fleet of UAVs in a risky environment with the presence of obstacles. For this purpose, we have developed a directed multi-graph where, for each pair of targets (points of departure and return of the mission included), we calculate a series of shorter trajectories with different limits of risk – from the risk-free path to the riskiest path. We then use a Tabu Search with two tabu lists. Using these procedures, we have been able to produce routes for a fleet of UAVs that minimize the cost of the mission while respecting the limit of risk and avoiding obstacles. Tests are conducted on examples created on the basis of descriptions given by the Canadian Defense and, also on some instances of the CVRP (Capacitated Vehicle Routing Problem), those described by Christofides et Elion and those described

by Christofides, Mingozzi et Toth. The results are of very satisfactory since all trajectories are non-holonomic and the improvement of the objective, when compared to a simple constructive method, achieves in some cases between 10 % and 43 %. We have even obtained an improvement of 69 %, but on a poor solution generated by a greedy algorithm.

TABLE DES MATIÈRES

DÉDICACE III

REMERCIEMENTS IV

RÉSUMÉ V

ABSTRACT IX

TABLE DES MATIÈRES XIII

LISTE DES TABLEAUX XVII

LISTE DES FIGURES XIX

LISTE DES ANNEXES XXIII

INTRODUCTION 1

CHAPITRE 1 PRÉSENTATION DE LA PROBLÉMATIQUE 5

1.1 Motivation 5

1.2 Contexte 6

1.3 Les UAVs (avions sans pilote) 8

1.4 La mission 11

1.5 La problématique 12

1.6 Hypothèses de travail 14

CHAPITRE 2 REVUE DE LA LITTÉRATURE 17

2.1 Différentes modélisations : types de problèmes 18

2.1.1 Un seul UAV 18

2.1.2 Une flotte d'UAV, cas statique 20

2.1.3 Une flotte d'UAV, cas dynamique 22

2.1.4 Modélisation du terrain 23

2.1.5 Design de la trajectoire d'un UAV 26

2.1.6 Identification de la cible.....27

2.2 Différentes approches de résolution.....27

2.2.1 Plus Court Chemin Continu Non holonome dans un terrain avec des Obstacles et des zones de Risque (PCCC_NH_OR).....28

2.2.2 Définition du TSP et du VRP.....29

2.2.3 Les différentes variantes du VRP.....33

2.2.4 Les techniques de résolution 36

2.3 Conclusion 44

2.4 Notre démarche 45

CHAPITRE 3 PLUS COURT CHEMIN CONTINU NON HOLONOME DANS UN ENVIRONNEMENT À RISQUE ET EN PRÉSENCE D'OBSTACLES..... 48

3.1 Le problème du plus court chemin non holonome.....50

3.1.1 Le terrain, les obstacles et les dangers 50

3.1.2 L'UAV et les systèmes non holonomes.....52

3.1.3 Le problème 53

3.2 Modélisation et Formulations mathématiques 55

3.2.1 Notations 56

3.2.2 Les variables de décision 57

3.2.3 Formulation 1 : le risque dans la fonction objectif 57

3.2.4 Formulation 2 : le risque dans la contrainte.....59

3.2.5 Modélisation du terrain 62

3.3 Résolution et Algorithme 76

3.3.1 Génération du graphe orienté : graphe complet 76

3.3.2 Relaxation de la contrainte de non-holonomie : génération du Graphe incomplet.....81

3.3.3 Calcul du plus court chemin : l'algorithme de Dijkstra.....84

3.3.4 Calcul du plus court chemin avec risque minimum : l'algorithme de Dijkstra modifié 85

3.4 Résultats et analyse 87

3.4.1 Discrétisation du terrain versus graphe de visibilité 88

3.4.2 La comparaison des deux formulations.....90

3.4.3 La relaxation de la contrainte cinématique 94

3.5 Conclusion 99

CHAPITRE 4 PLANIFICATION DE MISSION D'UNE FLOTTE DE DRONES NON HOLONOMES DANS UN ENVIRONNEMENT À RISQUE ET EN PRÉSENCE D'OBSTACLES 100

4.1 Le problème : la planification d'une mission de surveillance.....101

4.1.1 Objectif du problème.....103

4.1.2 Contraintes du problème 104

4.1.3 Hypothèses de travail 106

4.2 Formulation et modélisation 106

4.3 Résolution : le multi-graphe et la recherche Tabou 107

4.3.1 Modélisation du terrain 108

4.3.2 La recherche Tabou.....109

4.3.3 Prétraitement : génération du multi-graphe MG 110

4.3.4 L'espace de travail et la représentation des solutions 115

4.3.5 La fonction d'évaluation 116

4.3.6 La solution Initiale 117

4.3.7 Le voisinage et les listes taboues 118

4.3.8 Le fonctionnement du tabou 119

4.4 Tests et résultats 120

4.4.1 Qualité et précisions des solutions 121

4.4.2 Analyse plus détaillée du comportement du Tabou (exemple Terrain 2) 123

4.4.3 Calibrage des paramètres Tabou 128

4.4.4 Évaluation des performances du Tabou sur des instances de CVRP 133

4.5 Conclusion 138

CHAPITRE 5 CONCLUSION ET TRAVAUX FUTURS 141

5.1 Les éléments de conclusion 142

5.2 Les contributions 143

5.3 Futures avenues de recherche 144

BIBLIOGRAPHIE 146

ANNEXES 161

LISTE DES TABLEAUX

Tableau 1 : Descriptions des terrains 88

Tableau 2 : Comparaison des temps de génartion des graphes par discrétisation versus graphe de visibilité. 89

Tableau 3 : Analyse détaillée du temps de génération du graphe pour le terrain 3 avec des obstacles non convexes et puis avec obstacles convexes 89

Tableau 4 : Comparaison des résultats obtenus sur les différents graphes des deux méthodes. 90

Tableau 5 : Résultat par les deux formulations pour le terrain Map06. On remarque que les deux formulations sont équivalentes. 91

Tableau 6 : Comparaison de la génération d'un graphe complet (avec la contrainte de non-holonomie) versus un graphe incomplet (sans la contrainte de non-holonomie). 95

Tableau 7 : Comparaison des deux modélisations (des deux graphes) avec la formulation 1 (risque dans l'objectif). 95

Tableau 8 : Comparaison des deux modélisations (des deux graphes) avec la formulation 2 (risque dans la contrainte). 96

Tableau 9 : Exemple d'informations stockées dans le multi-graphe entre deux nœuds donnés 115

Tableau 10 : Description des terrains 121

Tableau 11 : Paramètres du Tabou pour le terrain 2 scénario 4 124

Tableau 12 : Les temps de calcul pour le terrain 2. 125

Tableau 13 : Amélioration de l'objectif en fonction des itérations et des scénarios pour le terrain 2. 125

Tableau 14 : Résultats obtenus sur les instances de CVRP pour le scénario 1 et au bout de 500 itérations. 135

Tableau 15 : Résultats obtenus après 500 itérations pour les différentes instances de CVRP et pour les 4 scénarios. (Les résultats sont exprimés en pourcentage de la meilleure solution connue sauf pour ‘M-n200-k16’.). 136

Tableau 16 : Amélioration de l’objectif au bout de 500 itérations pour chaque instance et chaque scénario. 137

LISTE DES FIGURES

Figure 1 : Exemple de mission de reconnaissance. *Source : http://www.zmka.hu/tanszekek/ehc/konferencia/may/balogh.htm* 7

Figure 2 : Quelques exemple d'UAVs : a) Avion à voilure fixe; b) Quad rotor; c) Hélicoptères; d) Blimp. *Source : Betancourt, A. S. J. (2009).* 8

Figure 3 : Exemple d'UAV. *Source : http://www.aaicorp.com/defense/uav/400.html* 10

Figure 4 : Modélisation du terrain 19

Figure 5 : Diagramme de Voronoï. *Source : http://amin.unice.fr/~clerel/TER/rapport/node6.html* 25

Figure 6 : Exemple de graphe de visibilité 26

Figure 7 : Exemple du problème du voyageur de commerce avec un seul dépôt. 31

Figure 8 : une solution possible avec deux véhicules pour le problème de la Figure 7… 31

Figure 9 : Notre démarche 47

Figure 10 : Exemple de terrain (les polygones représentent les obstacles et les cercles les zones de danger). 51

Figure 11 : Exemple de terrain avec des obstacles et des zones de danger. 61

Figure 12 : Le graphe correspondant au terrain de la Figure 11 et sur lequel le plus court chemin sera calculé. 61

Figure 13 : Un exemple de graphe généré par maillage. Les points encerclés sont des exemples de points intermédiaires non utiles 64

Figure 14 : Exemples de chemins réalisables (trait plein) qui ne traversent aucun obstacle et qui donnent donc naissance à un arc; exemples aussi de chemins non réalisables (en pointillé) qui traversent des obstacles et qui seront donc exclu du graphe. 65

Figure 15 : Exemples de sommets d'obstacles. a est un sommet illustrant les directions que l'avion ne peut avoir à son voisinage. b est un sommet que l'avion ne peut visiter et qui est donc exclu du graphe. 67

Figure 16 : Exemple de contournement des zones de danger sur un terrain où le diagramme de Voronoï est dessiné. Même le diagramme de Voronoï (traits en pointillé) ne permet pas le contournement des zones de danger. 68

Figure 17 : Illustration de l'intégration des zones de danger. Les points sur la circonférence représentent la discrétisation du contour de la zone de danger 69

Figure 18 : Les six courbes de Dubins. 70

Figure 19 : Implémentation des courbes de Dubins. (a : sans obstacles; b : le résultat des courbes de Dubins propose une courbe qui traverse l'obstacle; c : notre implémentation évite l'obstacle). 72

Figure 20 : Exemples de courbes de Dubins réalisables et non réalisables : a) la courbe LSL; b) la courbe RSL c) 73

Figure 21 : Illustration du calcul du risque. 75

Figure 22 : Les graphes associés au terrain de la Figure 10 (a : le graphe complet, 771 nœuds et 3987 arcs; b : le graphe incomplet, 81 nœuds et 502 arcs) 78

Figure 23 : Exemple de trajectoire holonome (issue du calcul sur le graphe incomplet, sans contrainte cinématique) 84

Figure 24 : La trajectoire corrigée, non holonome, correspondant à la correction de celle de la Figure 23. 85

Figure 25 : Trajectoire optimale obtenue sur le terrain Map06 selon les deux formulations. 92

Figure 26 : Trajectoire optimale obtenue sur le terrain Map06 selon les deux formulations et en utilisant un graphe incomplet (i.e. en relaxant la contrainte cinématique). 98

Figure 27 : Exemple de terrain où une mission de surveillance est à planifier.....102

Figure 28 : Exemple de deux trajectoires : une qui consomme plus de risque(en trait pointillé) et une qui consomme plus d'autonomie(en trait plein). 105

Figure 29 : Graphe orienté correspondant problème non holonome du terrain de la Figure 27. 108

Figure 30 : Exemples de routes visitant les mêmes cibles mais avec des niveaux de risque différents et avec différentes autonomies.....111

Figure 31 : Exemple de l'implication de l'insertion d'une nouvelle cible dans une route. 112

Figure 32 : Le multi-graphe associé au graphe de la Figure 29.....114

Figure 33 : Terrain 1, solution avec autonomie de 200 et risque max de 15. 122

Figure 34 : Terrain 1, solution avec autonomie de 250 et risque max de 0. 122

Figure 35 : Détail de la Figure 34 : une partie de la route 1 avec un zoom. 123

Figure 36 : Solutions initiale et finale pour le terrain 2 avec scénario 4.....126

Figure 37 : Exemple de résolution sur le terrain 2, en partant d'une solution (générée par un algorithme glouton).....127

Figure 38 : Évolution de l'objectif en fonction des itérations pour différentes tailles de liste taboue inter-route.....129

Figure 39 : Évolution de l'objectif en fonction des itérations pour différentes tailles de liste taboue intra-route.....129

Figure 40 : Évolution de l'objectif en fonction des itérations pour différents degré de finesse dans le multi-graphe (nombre d'arcs entre chaque paire de nœuds).....130

Figure 41 : Évolution de l'objectif en fonction des itérations pour différents degré de finesse dans le multi-graphe qui est généré selon la formulation 1 (voir chapitre 3). 132

Figure 42 : Terrain 'Map01'.....162

Figure 43 : Terrain 'Map02'.....162

Figure 44 : Terrain 'Map03'.....163

Figure 45 : Terrain 'Map04'.....163

Figure 46 : Terrain 'Map05'.....164

Figure 47 : Terrain 'Map06'.....164

Figure 48 : Terrain 'Map07'.....165

Figure 49 : Terrain 0 (3 obstacles, 2 zones de danger et 10 cibles).....166

Figure 50 : Terrain 1 (3 obstacles, 2 zones de danger et 40 cibles).....167

Figure 51 : Le terrain 2 (4 obstacles, 3 zones de danger et 20 cibles).....168

LISTE DES ANNEXES

Annexe 1 : L'implémentation de la modélisation du terrain 161

Annexe 2 : Les terrains utilisés pour le plus court chemin 162

Annexe 3 : Les terrains utilisés pour la planification de la mission..... 166

Annexe 4 : Extrait des itérations du Tabou pour le terrain 2 scénario 1..... 169

Annexe 5 : Exemple de résultats obtenus sur les instances du CVRP 170

INTRODUCTION

Les UAVs (Unmanned Aerial Vehicle ou RPV Remotely Piloted Vehicle), qu'on nomme aussi drones, sont des avions sans pilote, très prometteurs pour les nations qui veulent avoir du leadership technologique et préserver leur sécurité à moindre risque. Ils sont essentiels pour améliorer la sécurité de la société, en raison de leur potentiel à exécuter des activités dangereuses et des tâches répétitives dans des environnements hasardeux, éloignés et dangereux.

Les militaires les utilisent pour des missions de reconnaissance ou de surveillance, ils recueillent des informations multiformes sur des objectifs au sol et transmettent leurs images ou autres données par liaisons électroniques. En plus de ces missions militaires, ils servent à d'autres fins civils, telles que : surveillance du littoral, surveillance des grands territoires – comme le Grand Nord Canadien, aider et faciliter des opérations de sauvetage… etc. Bien que les récentes avancées technologiques ont permis l'essor des ces drones et que le développement de systèmes sans pilote et leurs implémentations ont prouvé leur grand potentiel dans les deux types d'applications militaires et civiles, il reste que, tirer le plein potentiel de ces engins ne serait complet que lorsqu'ils peuvent fonctionner de façon autonome.

Pour atteindre cette autonomie plusieurs défis doivent être relevés. Parmi eux, la planification de la mission d'une flotte de drones autonomes dans des zones très souvent pleines d'obstacles et de menaces. Cette mission doit être menée à bien de manière à réaliser les tâches confiées sans compromettre la réussite de la mission. Le design et l'optimisation de trajectoires autonomes pour ces véhicules est un problème complexe et crucial dans la planification et le contrôle des missions des drones. Il suscite l'intérêt de plusieurs chercheurs dans les domaines de la robotique, des véhicules sous-marins et des véhicules aériens (Latombe, 1991; Xidias et Aspragathos, 2008; Chandler et Pachter 1998; Pereira, Kumar et Campos, 2008; Nigam, 2009).

C'est le défi de cette planification que nous allons relever dans cette thèse. C'est un grand défi puisque le terrain est jonché d'obstacles qui rendent le vol difficile. Les drones ont une autonomie de vol limitée. Ils ont aussi des contraintes cinématiques, ils ne peuvent s'arrêter sur place, ni faire de marche arrière ni négocier n'importe quel virage. Le terrain de la mission, quant à lui, est un milieu hostile, il est truffé de radars qui guettent nos avions. Il faudra donc trouver une solution pratique et réalisable qui tient compte de toutes ces conditions.

Nous avons relevé ce défi dans cette thèse ce qui nous a permis de faire plusieurs contributions qui s'inscrivent dans le rapprochement entre deux domaines : celui du « contrôle des systèmes ». (Cassandras, 2002; Parker, 2000; Koo et al., 1998; Paw, 2009; Pelosi, 2010…) et celui de la recherche opérationnelle, des métaheuristiques (Hertz, 2005a et 2005b; Archetti, Hertz et Speranza, 2007; Laporte, 1992a et 1992b; Crainic et Gendreau, 2002; Pedersen, Crainic et Madsen, 2009;). Nous avons appliqué

des outils de la recherche opérationnelle à des problèmes traditionnellement traités comme des problèmes de contrôle, et l'apport est significatif.

La première contribution est dans la modélisation du terrain. Une modélisation qui intègre toutes les contraintes réelles du problème (la non-holonomie, l'autonomie, le risque et les obstacles) dans un graphe orienté et qui minimise au maximum la taille de ce dernier. La modélisation est valide quelque soit le degré de précision cherché, contrairement aux méthodes basées sur le quadrillage.

La deuxième contribution est celle reliée au calcul du « Plus court chemin continu non holonome dans un environnement à risque et en présence d'obstacles ». Le travail va permettre la soumission d'un article dans un proche avenir.

La troisième contribution est celle formulée par la réponse pratique à notre problème de « Planification de mission d'une flotte de drones non holonomes un environnement à risque et en présence d'obstacles ». Ce travail fera, aussi, l'objet d'un article à soumettre.

La présente thèse s'articule comme suit : dans un premier lieu, nous présentons la problématique générale (chapitre 1) pour esquisser l'étendue, la complexité et les enjeux de cette tâche dans le contexte de l'armée canadienne. Ensuite, le chapitre 2 sera réservé pour une revue de littérature non exhaustive mais assez large pour couvrir les différents aspects de la problématique. Ce chapitre sera conclu par la définition du problème spécifique qui nous concerne dans cet ouvrage et la présentation de notre démarche. Dans le chapitre 3, nous traitons un problème crucial pour notre planification de la

mission, c’est celui du « Plus court chemin continu non holonome dans un environnement à risque et en présence d’obstacles » (PCCC_NH_OR). La réponse à ce problème est constituée d'une modélisation du terrain en un graphe orienté intégrant toutes les contraintes et de deux algorithmes de calcul de plus court chemin sur un graphe. Le premier est une adaptation de l’algorithme de Dijkstra (1959) classique et le deuxième est une adaptation des travaux de Beasley et Christofides (1989) et Naghi et Soumis (2000) sur le calcul d'un plus court chemin avec des contraintes de ressources. Notre approche est novatrice car c'est la première fois qu'on traite ce problème dans sa totalité. Une fois ce plus court chemin déterminé, dans le chapitre 4, nous détaillerons notre solution pour la problématique de « Planification de mission d’une flotte de drones non holonomes un environnement à risque et en présence d’obstacles ». Nous formulons cette problématique comme un problème de tournées de véhicules avec capacités (CVRP), nous utilisons les résultats sur le plus court chemin PCCC_NH_OR puis la résolution se fait par une métaheuristique Tabou avec une double liste taboue. Ce travail fera aussi l’objet d’un article à soumettre. Finalement, le dernier chapitre est une conclusion de l’ouvrage où nous présentons les extensions possibles du présent travail et les éléments qui restent à développer.

CHAPITRE 1

PRÉSENTATION DE LA PROBLÉMATIQUE

Les véhicules aériens sans pilote (UAV) sont largement utilisés dans le monde entier pour diverses applications tant civiles que militaires. Dans le présent chapitre nous allons décrire les caractéristiques de ces avions et expliquer la problématique relative aux missions pour lesquelles l'Armée canadienne les utilise.

Premièrement, nous commençons par expliquer l'origine et la motivation derrière le projet de cette thèse. Ensuite, une mise en contexte suivra. Puis nous ferons le tour de tous les éléments de la problématique pour mieux les cerner. Ces derniers éléments sont : les avions, leurs caractéristiques et leur cinématique, la définition des cibles et leur identification, le terrain avec ces obstacles et les zones de danger et de surveillance radar. Finalement, nous concluons ce chapitre par l'énoncé de la problématique.

1.1 Motivation

Les UAVs (Unmanned Aerial Vehicle), avions sans pilote ou drones, sont l'objet d'un intérêt croissant de la part de plusieurs pays pour les aider dans leurs missions militaires. Dans le cas de notre projet, les forces armées canadiennes sont intéressées par ce type d'engins dans le cadre de leurs missions de paix à travers le monde. La défense

canadienne a présentement recours aux UAVs pour protéger certains territoires et ses troupes dans leur mission en Afghanistan. Ce n'est toutefois qu'un début. Le Canada veut devenir un plus grand expert dans l'utilisation des UAVs. Le but de ce projet est d'initier, voire développer l'expertise de surveillance et / ou d'exploration d'un territoire avec plusieurs UAVs à la fois.

Un autre intérêt pour le Canada derrière ce projet est que ses applications sont aussi grandes dans le civil que dans le militaire. Le Canada est un pays vaste avec de grandes forêts et un large littoral. L'usage de ces flottes d'UAVs faciliterait grandement la surveillance de ces territoires et de ce littoral. Les missions de sauvetage en forêt, la surveillance des feux de forêt, le contrôle du littoral et de la frontière seront plus simples et coûteraient moins cher avec une flotte d'avions sans pilote totalement autonome. De plus, les résultats de cette recherche sont applicables à n'importe quelle autre flotte de véhicules robotisés tels que les robots sous-marins ou les robots sur roues.

1.2 Contexte

Les militaires utilisent les UAVs pour deux types d'opérations : soit la reconnaissance d'un territoire hostile où il est risqué d'envoyer des soldats, soit pour la surveillance d'une zone dont on a le contrôle pour des opérations de maintien de la paix. Dans les deux cas, l'UAV doit se déplacer dans un milieu hostile pour visiter des cibles prédéterminées.

La Figure 1 illustre le cas de reconnaissance d'un terrain ennemi par un seul UAV. Ce cas on n'a pas le contrôle de la zone ennemie à visiter et on dispose de peu

d'informations sur cette dernière. Les troupes menant la mission se trouvent à gauche de la figure et l'ennemi se trouve à droite de la frontière en pointillée. Avant de s'engager aveuglément dans le terrain hostile, les soldats envoient un UAV qui est catapulté du point « take off » sur la Figure 1 et suit une trajectoire qui lui permettra de balayer stratégiquement le terrain ennemi. Durant son exploration, l'UAV est en contact – radio – continu avec le poste « ground terminal » qui collecte les informations. Cette collecte de données permet aux soldats de mieux planifier leurs futures attaques de ce territoire.

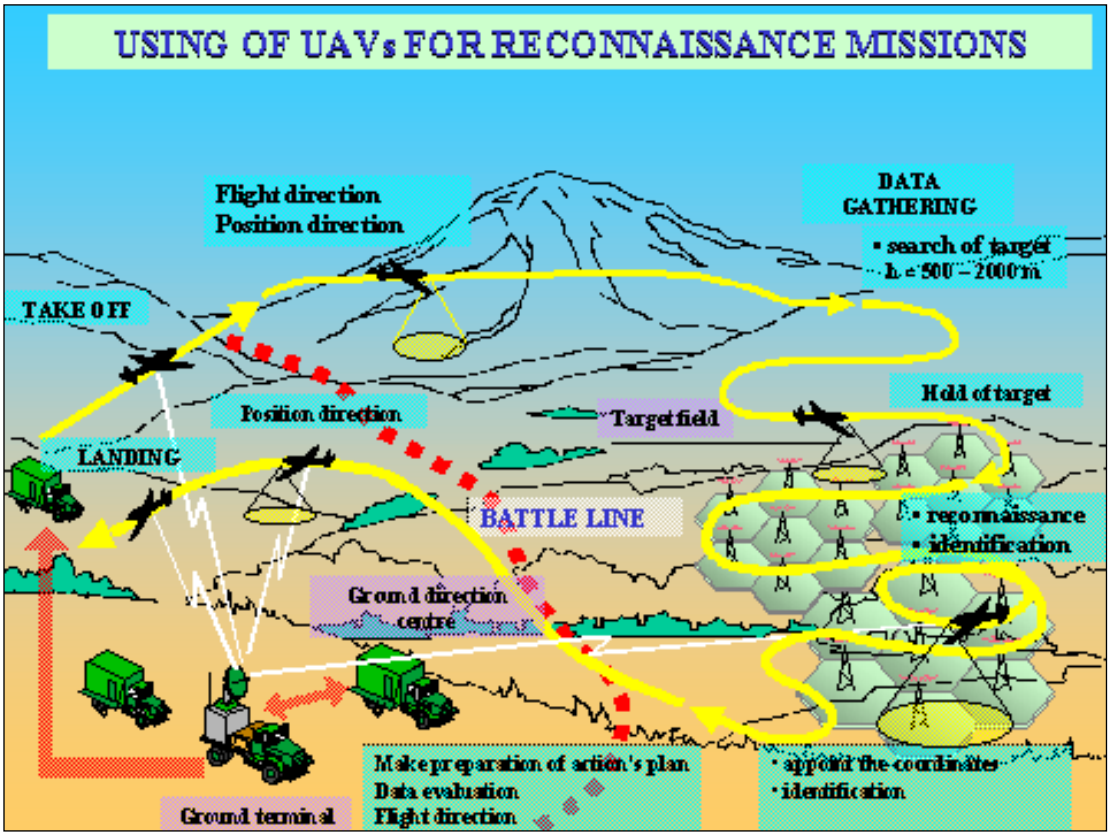


Figure 1 : Exemple de mission de reconnaissance.
Source : http://www.zmka.hu/tanszekek/ehc/konferencia/may/balogh.htm

1.3 Les UAVs (avions sans pilote)

Les UAVs sont des avions sans pilote utilisés pour des missions de reconnaissance ou de surveillance, ils recueillent des informations multiformes sur des objectifs au sol et transmettent leurs images ou autres données par liaisons électroniques.

Figure 2 : Quelques exemples d'UAVs : a) Avion à voilure fixe; b) Quad rotor; c) Hélicoptères; d) Blimp. *Source : Betancourt, A. S. J. (2009).*

La Figure 2, nous montre quelques types d'UAVs qui sont utilisés dans le monde. Il en existe de toutes sortes et de différentes formes : des avions à ailes fixes, des hélicoptères, des dirigeables… etc. Betancourt (2009) et Paw (2009), dans leurs travaux présentent en détail ces engins et leurs systèmes de contrôle. Dans notre projet on ne s'intéresse pas au

contrôle de ces avions, mais nous devons comprendre comment les commandes sont transmises.

La Figure 3 nous montre un exemple d'avion sans pilote. Il s'agit du « Shadow 400 » un type d'UAV développé par la marine américaine pour des missions de surveillance, reconnaissance et d'observation dont le but principal est d'appuyer les opérations navales. On voit sur la figure les différentes caractéristiques de cet engin qui nous donnent une idée sur ces capacités : une vitesse entre 120 et 185 km/h, une autonomie de 5 heures, un poids de 201 kg et une envergure de 3.82 m x 5.1 m. Il est aussi équipé de caméras vidéo et d'un système de télécommunication supplémentaire – pour pallier aux défaillances du lien de communication. Le « Shadow 400 » peut être lancé à partir d'un navire ou à partir du sol et guidé à distance.

Comme pour le « Shadow 400 », les UAVs sont de taille compacte, et de petit poids, ce qui fait d'eux des appareils idéaux pour mener des missions de surveillance ou de reconnaissance autrement très coûteuses en termes de ressources. Le fait qu'ils soient sans pilote permet de mener des opérations à haut risque, et ce, sans pertes humaines éventuelles.

Performance

> Endurance: 5 hours
> Max Altitude: 12,000 ft. (3,660 m)
> Max Dash Speed: 100kts (185 kph)
> Cruise Speed: 75 kts (139 kph)
> Loiter Speed: 65 kts (120 kph)

Features

> All Composite Construction
> Multiple Launch and Recovery Methods
> Shipboard Operations Qualified
> Airborne Video Tape Recorder
> Multiple Sensor Capability
> Dual Redundant Data Links
> Long-Life Emergency Battery

Technical Data

> Length: 12.54 ft. (3.82 m)
> Wingspan: 16.80 ft. (5.1) m
> Max Payload: 66 lbs. (30 kg)
> Empty Weight: 147 kg
> Max Gross Weight: 442 lbs. (201 kg)

Figure 3 : Exemple d'UAV.
Source : http://www.aaicorp.com/defense/uav/400.html

1.4 La mission

Une mission consiste en la visite, par une flotte ou un seul UAV, de plusieurs cibles pour collecter des informations de type image. Chaque cible est définie par ses coordonnées géographiques x et y et par l'angle θ selon lequel la photo, ou l'information, doit être collectée. Un point géographique peut représenter plusieurs cibles si la collecte de données nécessite plusieurs visites selon différents angles. Les missions peuvent être de type surveillance – d'une zone sous contrôle pour des opérations de maintien de la paix – la majorité des cibles est connue d'avance, ou de type reconnaissance – d'un territoire hostile où il est risqué d'envoyer des soldats et où on découvre les cibles au fur et à mesure que la mission se déroule. Dans le contexte canadien, par opposition à celui américain, on a recours à ces engins surtout pour des missions de surveillance. L'armée canadienne utilise ces UAVs surtout pour sécuriser un territoire ou tout simplement garder en sécurité ses troupes. La Figure 1 illustre un cas de reconnaissance.

Dans le contexte de surveillance d'un terrain donné, nous sommes en présence d'une situation identique à celle de la Figure 1 excepté que les points « chauds » à surveiller sont connus et que les soldats doivent les garder en toute sécurité. Or, comme on le voit bien sûr la Figure 1, un UAV est incapable de couvrir d'énormes étendues : son autonomie de quelques heures est limitée, sa vitesse de l'ordre de 120 km/h et son altitude maximale de 4000 m. Un seul UAV est incapable de mener toute une mission de surveillance en un temps raisonnable. De plus, il est vulnérable : il ne peut pas échapper aux radars et tirs ennemis par sa basse altitude. On a donc recours à une flotte d'UAVs qui, en se partageant la zone, permet une couverture en un temps acceptable. De plus,

l'ensemble d'avions est moins vulnérable. C'est-à-dire la mission ne sera pas compromise avec une flotte d'avions car si un avion est repéré par l'ennemi, on peut redéployer les autres avions à tout moment et assurer la réussite de la mission. Jusqu'à présent, le contrôle d'UAVs ou de cette flotte se fait manuellement. C'est un opérateur qui guide les avions selon un plan préétabli.

Toutefois, cette tâche manuelle est très ardue. En effet l'environnement très hostile et dynamique rend la coordination et le contrôle de cette flotte très complexe. On doit tenir compte à la fois de la géographie du terrain, des conditions météorologiques et des dangers potentiels qui guettent les avions tels que les attaques ennemies et les radars adversaires. À tout moment, les avions risquent des attaques ennemies, des accidents ou des collisions avec les obstacles ou entre eux. En outre, à tout moment, il peut être question de redéployer ces avions et recalculer leurs trajectoires pour pallier aux aléas. Lorsque l'on augmente le nombre d'UAVs, on constate rapidement qu'on a besoin d'un support informatique pour faciliter les opérations de surveillance.

1.5 La problématique

La question qui se pose alors est la suivante : **peut-on concevoir un système de contrôle en temps réel pour coordonner une flotte d'UAVs en mission de surveillance ou de reconnaissance d'un territoire?**

La problématique est de développer une stratégie de déploiement (et / ou redéploiement) d'une flotte d'UAVs dont la mission est d'explorer un territoire hostile. Cette stratégie doit assurer :

- que la durée de la mission soit la plus courte possible puisqu'on est dans un territoire ennemi (dans le cas de notre thèse);
- la protection des UAVs (assurer leur survie durant la mission) car leur élimination est synonyme de l'échec de la mission;
- la transmission des informations au quartier général (QG) – c'est le but principal;
- (aussi) la communication entre les drones pour permettre une meilleure collaboration;
- la stratégie elle-même doit être très réactive car le contexte est dynamique et on opère en temps réel.

Certains objectifs associés à notre problématique :

- assurer la visite des cibles connues fixes ou mouvantes;
- minimiser la durée de la mission;
- maximiser le nombre de cibles visitées;
- maximiser la couverture du terrain;
- pallier à la défaillance ou la perte d'une partie de la flotte (via le redéploiement de la flotte);
- collecte d'informations (généralement images);
- assurer la transmission des informations au QG;
- minimiser le nombre de communications entre UAVs;
- minimiser les communications avec le QG.

Au sein de toutes ces variantes, la problématique qui nous intéresse se situe comme suit : on a un nombre fixe d'UAVs et on cherchera à planifier leur mission de surveillance d'un terrain, plein d'obstacles et de zones de danger, en vue de visiter toutes les cibles connues d'avance. L'objectif est de minimiser le coût et la durée de la mission tout en gardant le niveau de risque inférieur à un certain seuil et en respectant les contraintes des véhicules. L'objectif général de notre problème consiste, donc, à assurer la couverture, la visite des points chauds, d'un territoire donné par une flotte d'UAVs donnée (nombre d'UAVs fixe) lors d'une mission de surveillance, en visitant des points clés sous des contraintes de conditions météorologiques, géographiques, de danger (présence d'ennemis), d'autonomie et de manœuvrabilité des UAVs.

1.6 Hypothèses de travail

Pour réaliser notre projet on considère les hypothèses suivantes :

- les vols se font à altitude constante et à vitesse constante;
- le problème est en trois dimensions, mais vu que les vols se font à une altitude constante, on va modéliser le terrain en deux dimensions uniquement;
- une configuration du véhicule est définie par les coordonnées (x, y) de l'avion et l'orientation de ce dernier qui sera modélisée par un angle θ;
- les obstacles sont définis par des polygones convexes ou non convexes et on suppose que les avions peuvent longer ces derniers, c'est-à-dire que les trajectoires des drones peuvent être tangentes aux obstacles;

- les zones de dangers sont définies par des disques dont le centre et le rayon sont donnés. On considère que le radar a une portée limitée définie par ce rayon;
- la non-holonomie de la trajectoire pour notre cas, celui des drones, est définie par la contrainte cinématique, ou de manœuvrabilité de l'avion reliée à son angle limite de braquage. Dans la suite elle sera caractérisée par la donnée de cet angle maximum ou par la donnée d'un rayon de courbure minimum;
- tous les virages des avions se font avec un même rayon fixe, le rayon minimum de courbure. Cette hypothèse est toujours valide et n'enlève rien à la qualité de notre solution. En effet, en termes de longueur de la trajectoire, Dubins (1957) a démontré que ce rayon minimum est celui qui garantit la trajectoire la plus courte. Deuxièmement, on pourrait prétexter qu'un rayon plus grand permettrait de contourner certains obstacles que le rayon minimum ne permettrait pas. Ceci est faux, car dans notre modélisation nous tenons compte de tous les sommets des obstacles ce qui permet de contourner tous les obstacles et à moindre coût, avec une longueur de trajectoire plus petite. Donc cette hypothèse est, en fait, une condition garantissant la plus courte trajectoire;
- il s'agit dans notre cas de drones, et il n'y a donc pas de marche arrière permise.

Pour répondre à cette problématique, on commence dans le chapitre qui suit, par une revue de la littérature afin d'investiguer les travaux déjà effectués dans ce domaine et de définir notre contribution. Le but est de prouver que cette question n'est pas encore totalement traitée.

CHAPITRE 2

REVUE DE LA LITTÉRATURE

Dans la littérature, on constate l'existence de différentes variantes à notre problème, différentes approches de résolution et plusieurs domaines connexes qui partagent la même problématique.

La revue de la littérature se divise en trois parties : la première concerne les différentes modélisations abordées dans la littérature, à savoir les différents sous-problèmes liés à la gestion des UAVs en commençant par les cas les plus simples et les différents efforts de modélisation du problème global. La deuxième partie se penche plutôt sur les différentes approches de résolution, soit des méthodes exactes, des heuristiques, des métaheuristiques ou méthodes hybrides s'inspirant de différentes disciplines connexes. Puis, dans la dernière partie, on s'intéresse aux situations dynamiques où des solutions doivent être apportées en temps réel.

Dans notre revue de la littérature, nous ne traitons pas l’aspect « contrôle des systèmes ». Les articles de Paw (2009), Betancourt (2009), Cassandras et Li (2002), Schoenwald (2000), Parker (2000) et Koo et al. (1998), présentent une introduction aux problèmes techniques et matériels reliés au contrôle. Le groupe de recherche en contrôle des

systèmes de la défense canadienne à Valcartier s'occupe de ce volet de recherche dans le projet sur les UAVs.

2.1 Différentes modélisations : types de problèmes

La coordination d'une flotte d'UAVs est un problème très vaste et très complexe. Il touche à plusieurs aspects : modélisation du territoire à surveiller, affectation des cibles aux UAVs, calcul de la trajectoire de chacun des UAVs, etc. Dans cette section, nous examinons, dans un premier lieu, le cas le plus simple, celui d'un seul UAV. Ensuite, nous présentons le cas de plusieurs UAVs. On s'attarde sur les détails de la modélisation du terrain et du calcul de trajectoires de vol des UAVs.

2.1.1 Un seul UAV

Le problème de base est celui où on considère un seul UAV devant visiter une cible donnée sous des contraintes de dangers et d'obstacles. Koo et al. (1998, 1999, 2001) se sont intéressés à ce problème. La Figure 4 donne une représentation d'un tel problème : les zones foncées (rouges) indiquent les dangers potentiels, les zones grises les obstacles, le carré noir (O) est l'origine de lancement de l'UAV, le carré noir (C) la cible à visiter et les autres points noirs sont les points intermédiaires « Way-points» par lesquels l'UAV peut passer. Il s'agit d'un problème du plus court chemin entre une origine (O) et une destination (C) sur un graphe donné, un problème très classique.

Gary (2000) présente un cas plus complexe, soit celui d'un seul UAV avec plusieurs cibles à visiter. Toutes les données sont connues a priori et ne changent pas durant la mission. Cette modélisation est celle du problème du voyageur de commerce TSP

(Traveling Salesman Problem). Dans ce cas on a deux types d'objectifs : (1) soit on cherche à minimiser le coût de couverture, il faut visiter toutes les cibles en minimisant le coût de la mission; (2) soit on cherche à visiter le plus de cibles si on a un temps maximal pour la mission.

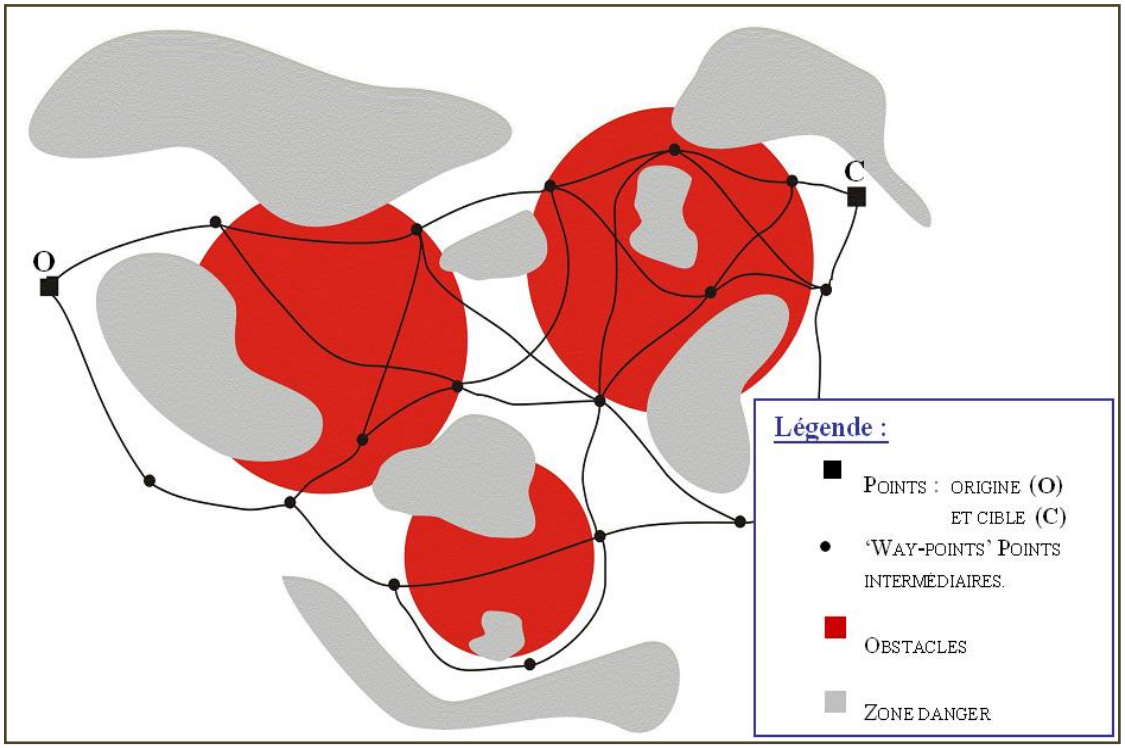


Figure 4 : Modélisation du terrain
Source : The Laboratory for Information and Decision Systems, MIT.

Dans la réalité, un seul UAV n'est pas suffisant. Il est toutefois pertinent de débuter la modélisation par un seul UAV avant d'attaquer le problème d'une flotte d'UAVs. De plus, le problème statique peut s'avérer intéressant avant d'attaquer le cas dynamique. Les travaux effectués sur un seul UAV sont intéressants mais limités. Ils ne peuvent pas être exploités pour résoudre de larges problèmes tels que celui de la planification de

mission d'une flotte de drones. Il y a beaucoup à faire pour supporter la base scientifique des autres modèles de contextes plus complexes.

2.1.2 Une flotte d'UAV, cas statique

À cause des limitations techniques d'un seul UAV, on a recours à une flotte, un ensemble de plusieurs UAVs qu'il faut déployer au-dessus du terrain à surveiller. Dans ce cas, on a plusieurs problématiques : (1) soit on cherche à minimiser le nombre d'UAVs nécessaire pour couvrir le terrain (Ryan et al., 1999; Sisson, 1997; Gary, 2000). (2) soit, pour une flotte donnée d'UAVs, on cherche à minimiser le temps total nécessaire pour la visite de toutes les cibles (minimiser le coût) ou de maximiser le nombre de cibles visitées tout en tenant compte de leur importance relative (Gary, 2000). Bellingham (2002) et Kuwata (2003) présentent un compromis entre ces deux objectifs. Nous élaborons sur ces deux approches.

Gary (2000) considère le cas de surveillance d'un territoire par une flotte d'UAVs dans son mémoire présenté à la « Air Force Institute Of Technology ». Il représente ce problème comme un cas de tournée de véhicules VRP (Vehicle Routing Problem) avec des contraintes de fenêtres de temps, dépôts multiples et une flotte hétérogène. Dans ce modèle, les véhicules sont les UAVs, les clients sont les cibles à visiter (voir points noirs sur la Figure 4) et l'entrepôt est le centre de commande terrestre d'où sont lancés les UAVs. Il considère le territoire à surveiller en deux dimensions, les distances entre les cibles étant connues. Le problème est défini par un graphe dont les nœuds sont les cibles et les arcs représentent les distances entre les différentes paires de nœuds. Gary s'est

inspiré des travaux de Laporte (1992a, 1992b) et de Tsubakitani et Evans (1998) pour résoudre son modèle de VRP par une méthode hybride « Jump Search / Tabu Search ». Il construit une liste de solutions initiales (Jump list) qu'il améliore par une méthode Tabou. Les travaux restent limités à des cas statiques.

Bellingham (2002) et Kuwata (2003) considèrent le problème de coordination d'une flotte d'UAVs qui doit explorer un terrain défini en deux dimensions avec des cibles prédéterminées et des points intermédiaires « Way-points» – qui sont les points des chemins les moins dangereux, calculés en se basant sur le diagramme de Voronoï où les points dangereux sont des cibles à éviter (voir section 2.1.4). Puis, Bellingham (2002) modélise cette situation comme un VRP qu'il résout en deux étapes : (1) un problème d'affectation : l'affectation des cibles aux UAVs sous des contraintes de capacité (autonomie), avec un objectif de minimisation du temps total de la mission et de maximisation de la réussite de la mission, c'est-à-dire en utilisant au maximum les chemins les moins dangereux. (2) design de trajectoires : détermination de la trajectoire de chaque UAV en utilisant un maximum de points de passages « Way-points» et calcul de la durée de la trajectoire. Le design de la trajectoire se fait sous les contraintes d'obstacles, de manœuvrabilité et de minimisation d'exposition aux dangers.

L'article de Sisson (1997) est l'un des rares à intégrer des conditions climatiques et les attaques ennemies dans le modèle. Il propose un algorithme RTS (Reactive Tabu Search) basé sur le travail de Carlton (1995). Il y intègre les effets du vent et la perte d'UAVs due aux actions ennemies, des effets qu'il a générés grâce à des simulations Monte-Carlo.

Les modélisations d'une flotte d'UAVs, rencontrées dans la littérature, restent limitées et ne couvrent pas toutes les facettes de la problématique. On s'est contenté de trouver une approche connue de la recherche opérationnelle et d'utiliser des algorithmes faciles à implanter. Avant d'attaquer des contextes plus complexes, il faut voir si on peut être plus ingénieux au niveau de la modélisation. Pour cela, nous nous sommes assurés, avec les militaires, des enjeux de la surveillance et de la reconnaissance du terrain.

2.1.3 Une flotte d'UAV, cas dynamique

Jusqu'à présent, dans les travaux qu'on a révisés, le terrain est statique, toutes les données du problème sont connues avant la résolution : on résout le problème puis on implante la solution sur le terrain. On ne tient pas compte du côté dynamique du problème réel (mouvement des cibles, changement des obstacles et des dangers, apparition de nouvelles informations, dangers ou cibles…). Dans cette section, on voit quelques approches pour tenir compte de cette dimension du problème.

Les premières approches dynamiques qui ont été proposées consistent à résoudre le problème statique chaque fois que le contexte change. Gary (2000) propose aux opérateurs des UAVs « AVOs » (Air Vehicle Operators) un outil d'aide à la décision à la fois rapide et efficace. Ainsi chaque fois qu'il y a un changement, l'AVO lance le même algorithme pour générer une nouvelle solution, ce qui permet de réagir au changement du contexte. Cependant, cette stratégie reste très limitée et le nombre d'exécutions est limité par la vitesse de calcul de l'ordinateur et la taille du problème.

Gary a surtout essayé de proposer des algorithmes rapides pour que l'AVO soit assez réactif.

O'Rourke (1999) est l'un des chercheurs à avoir vraiment pris en compte le contexte dynamique des UAVs, les changements des données durant l'exécution de la mission. Il se base sur l'algorithme RTS (Reactive Tabu Search) de Carlton (1995) pour développer une application java qui permet des routes dynamiques. Il s'agit d'un algorithme qui part de la localisation courante des UAVs et calcule le meilleur chemin de retour au dépôt (centre de commande) en se basant sur le RTS. Là encore, le temps de réponse dépend de la taille du problème. Si le degré de dynamisme du problème est élevé l'algorithme est incapable de fournir des solutions.

État donné que le temps de calcul des trajectoires dans le cas dynamique est très élevé, il faudra, d'une part, identifier les parties de calculs à faire en prétraitement et ainsi minimiser la charge de calcul durant l'exécution de la mission. Et d'autre part, trouver dans la littérature quelques approches algorithmiques qui sont plus pertinentes au contexte dynamique.

2.1.4 Modélisation du terrain

Dans les travaux rencontrés durant la revue de la littérature, les terrains sont modélisés en deux dimensions selon une vue aérienne (projection vue de dessus) sans altitude. Comme dans la Figure 4, le terrain ainsi que les obstacles et les zones de danger sont représentés selon leurs coordonnées dans le plan et sans tenir compte de leur élévation. Tenir compte de cette élévation n'est pas nécessaire dans notre cas où les vols se font à

altitude constante. De plus, ce niveau de sophistication augmente grandement les calculs des trajectoires, distances et temps de parcours.

Une trajectoire de drone réalisable peut passer par n'importe quel point du terrain. Or planifier une trajectoire en considérant tous les points du plan est un problème non réalisable car on ne peut dénombrer tous les points du plan. Alors Dans la littérature, pour la modélisation du terrain, on a recours à la discrétisation de l'espace de configurations ou l'introduction de points de passage intermédiaires « Way-points» (Bellingham et al., 2001; Kuwata, 2003). Il existe trois façons de procéder :

Soit on utilise des grilles d'exploration : le maillage consiste à découper le terrain en petites mailles de taille identique ou non (De Mot et al., 2002). Le réseau ainsi obtenu forme le graphe (Les sommets du maillage sont les nœuds du graphe et les segments du maillage forment les arcs) sur lequel on résout un problème classique de plus court chemin. Ceci permet d'éviter les obstacles mais la précision de la méthode dépend de la finesse du maillage laquelle, si elle est petite, augmente grandement la taille du problème.

Soit on se base sur le diagramme de Voronoï (Bortoff, 2000). Ce diagramme est un découpage du territoire où les frontières représentent les lignes équidistantes de points localisés sur un plan. Dans notre contexte, les points sont les endroits les plus dangereux et le graphe est l'ensemble des chemins les moins dangereux. Pour simplifier, on utilise seulement les points des jonctions du diagramme de Voronoï, qui deviennent alors les points intermédiaires ou « Way-points ». Dans la Figure 5, nous avons un exemple de

diagramme où on peut considérer les points noirs comme étant les radars ennemis et le réseau d'arcs est le diagramme de Voronoï. Chandler (2001) utilise ce diagramme pour le calcul de la trajectoire où il cherche à minimiser le temps d'exposition aux radars ennemis. Cette méthode permet de tenir compte des zones de danger mais pas des obstacles.

Soit on utilise le graphe de visibilité (Richards et al., 2002) qui permet de réduire grandement la taille du graphe (voir Figure 6, ceci sera détaillé plus loin).

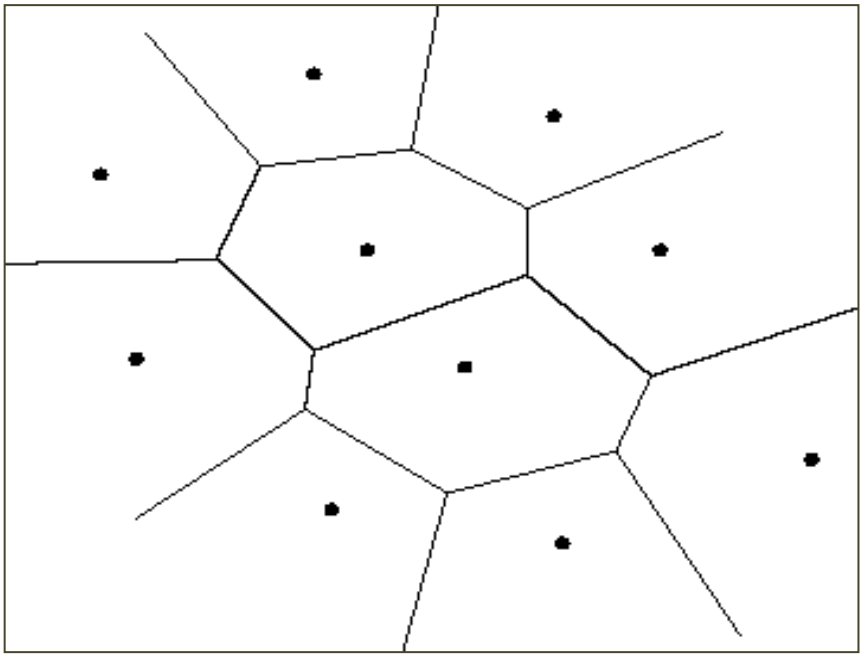


Figure 5 : Diagramme de Voronoï.
Source : http://amin.unice.fr/~clerel/TER/rapport/node6.html

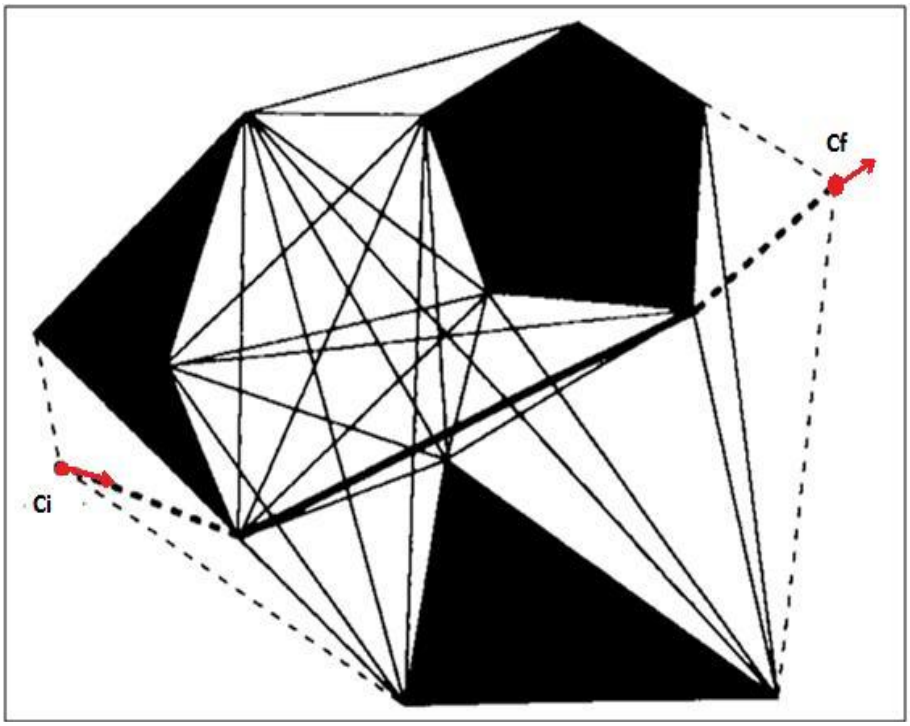


Figure 6 : Exemple de graphe de visibilité

2.1.5 Design de la trajectoire d'un UAV

Pour le design des trajectoires, on se base généralement sur un terrain en deux dimensions modélisé grâce à un diagramme de Voronoï. Dans la revue de la littérature toutes les trajectoires sont déterminées à hauteur constante. Le vol d'oiseau entre les différents Way-points est utilisé mais plusieurs tiennent compte des contraintes de manœuvrabilité d'un avion.

Chandler (2001) s'est intéressé au problème de détermination des trajectoires des UAVs dans un terrain en deux dimensions. Pour le calcul de la trajectoire, Chandler cherche à minimiser le temps d'exposition aux radars ennemis en se basant sur le digramme de Voronoï avec des trajectoires planes et en vol d'oiseau. Sigurd et How (2003) se sont intéressés au problème de collision entre les avions dans le cas de coordination d'une

flotte d'UAVs. Ce problème est présent, surtout quand la taille de la flotte est grande. Là aussi le problème est modélisé avec un terrain en deux dimensions et les cibles sont connues d'avance. Cependant pour répondre au problème de collision, Sigurd et How (2003) ont eu recours à des analogies avec les champs magnétiques. C'est tout de même un contexte simple où l'on suppose que les UAVs volent tous à la même altitude et qu'on ne peut pas prévoir le temps de passage des avions à certains points. Reste que le problème de collision est un facteur dont il faut tenir compte lorsqu'on utilise une flotte d'UAVs.

2.1.6 Identification de la cible

Chandler (2001) s'est intéressé au problème d'identification des cibles. Il a essayé de répondre à la question suivante : combien de photographies sont suffisantes pour identifier la cible et à quels angles doivent-elles être prises? Pour répondre à cette question, il s'est basé sur une distribution probabiliste qui décrit la qualité de l'image selon la distance entre l'UAV et la cible et selon l'angle de prise de vue par rapport à la plus grande dimension de la cible. Cet aspect est important, car il implique un repositionnement adéquat de l'avion pour avoir les angles de photos adéquats.

2.2 Différentes approches de résolution

Les problèmes de base qui concernent notre problématique sont principalement : le problème du voyageur de commerce TSP et le problème de tournées de véhicules VRP. Dans tous les travaux, le principe de base est la modélisation du terrain sous forme d'un graphe suivi de l'application d'un algorithme ou heuristique de calcul de plus court

chemin. Sur ce graphe le problème est modélisé en un problème de type VRP dans le cas d'une flotte d'UAVs ou du type TSP dans le cas d'un seul UAV. Un autre problème aussi important et commun à tous les cas ci-dessus est celui du « Plus Court Chemin Continu Non holonome dans un terrain avec des Obstacles et des zones de Risque (PCCC_NH_OR) ».

Ainsi dans cette section on va faire, premièrement, le tour de différentes méthodes connues pour la résolution de ces trois types de problèmes : on commence par le PCCC_NH_OR, puis les problèmes de TSP et VRP.

2.2.1 Plus Court Chemin Continu Non holonome dans un terrain avec des Obstacles et des zones de Risque (PCCC_NH_OR)

Après la modélisation du terrain en termes de graphe, on utilise des algorithmes pour le calcul du plus court chemin. Toutefois, aucune de ces méthodes ne couvre le problème en totalité avec toutes les contraintes. Il existe des travaux concernant une ou deux des contraintes :

- plus court chemin non holonome;
- plus court chemin avec obstacles;
- plus court chemin avec risque minimum;

Dubins (1957) est le premier à traiter le problème de chemin non holonome en introduisant les courbes qui portent son nom. Reeds et Shepp (1990) et Desaulniers et al.

(1995) ont introduit d'autres courbes et ont pu étendre le travail de Dubins à d'autres types de robots et tenir compte des obstacles.

Pour le problème du plus court chemin avec risque minimum, Kim et Hespanha (2003) introduisent un algorithme qui permet de calculer le plus court chemin dans un milieu anisotrope en pondérant les longueurs des arcs d'un graphe obtenu par discrétisation.

La majorité des travaux partent d'un graphe, qui forme la donnée du problème et qui est obtenu par la discrétisation du terrain selon l'une des méthodes ci-dessus, puis essaient de tenir compte des autres contraintes au niveau du calcul du plus court chemin sans vraiment tenir compte du risque. Dans notre thèse, nous proposons une modélisation du terrain qui traite bien toutes les contraintes (évitement des obstacles, intégration des zones de danger et la non-holonomie) et ainsi facilite le calcul du plus court chemin et du risque associé à ce chemin.

2.2.2 Définition du TSP et du VRP

Le problème de construction de tournées de véhicules, VRP, est utilisé en premier pour modéliser de nombreuses situations réelles dans le contexte de systèmes de distributions (Dantzig et Ramser, 1959) puis ultérieurement, pour des situations plus générales telles que la coordination des UAVs (Bellingham, 2002; Gary, 2000; O'Rourke, 1999). La résolution de ce problème par des heuristiques et des métaheuristiques est un domaine très actif. Il existe beaucoup de travaux sur ces heuristiques (Laporte et Semet (2002); Gendreau et al. (2002); Cordeau et al. (2002a); Cordeau et Laporte (2004)).

Le problème de base de cette classe consiste en une flotte de K véhicules, $k = 1 \ldots K$, identiques ou non, de capacités Q_k qui livrent un certain produit à I clients, $i = 1 \ldots I$, à partir d'un ensemble de L dépôts, $l = 1 \ldots L$. La demande d'un client i est notée d_i et l'offre de chaque dépôt l est notée O_l. Le coût de transport d'un point i vers le point j est noté c_{ij}. Le problème peut être modélisé sous forme d'un graphe. La Figure 7 illustre le cas d'un VRP avec un seul dépôt. Le véhicule doit partir du dépôt, situé au centre de la figure et desservir les clients avant de revenir à ce même dépôt. Chaque client à une demande donnée et la capacité du véhicule est limitée, donc ce dernier ne pourra pas visiter tous les clients en un seul tour. Il faudra faire plusieurs tours avec le même véhicule ou avoir plusieurs véhicules et chacun fait un tour. Dans les deux cas nous devons déterminer plusieurs routes permettant de visiter tous les clients et sans dépasser la capacité du véhicule (ou des véhicules).

La solution de ce problème, donc, consiste à déterminer un ensemble R d'au plus K routes, K étant le nombre de véhicules disponibles. Chaque route commence par un dépôt, visite un certain nombre de clients et retourne au dépôt. Chaque client est affecté à une et une seule route, la somme des demandes des clients d'une même route est inférieure ou égale à la capacité du véhicule utilisé. La solution est optimale si la somme des coûts est minimale. La Figure 8, présente une solution possible au problème de la Figure 7. Cette solution est constituée de 3 routes. Chaque route démarre au dépôt indique les clients à visiter par le véhicule, sans dépasser sa capacité, et retourne au dépôt.

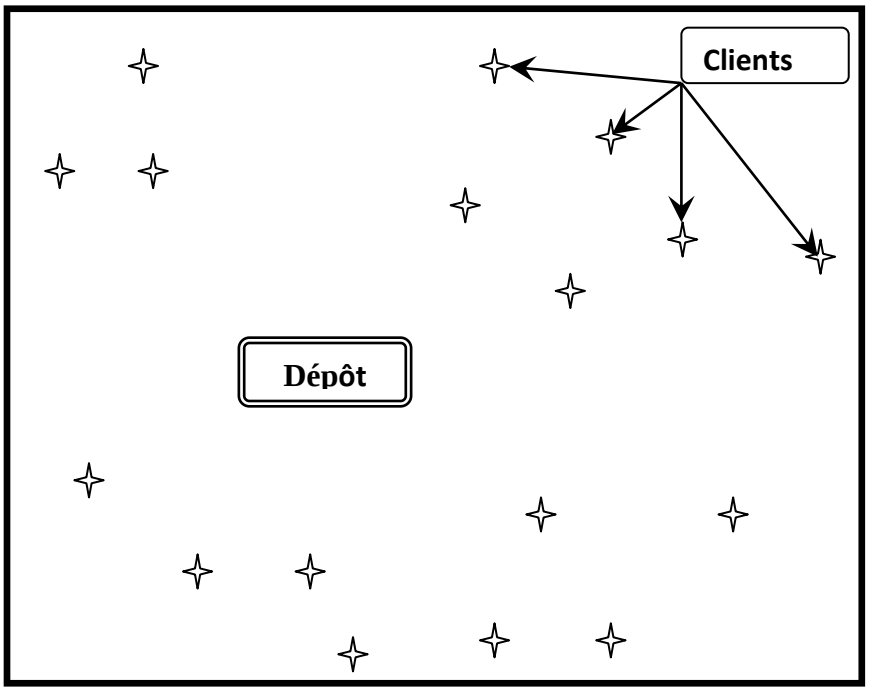


Figure 7 : Exemple du problème du voyageur de commerce avec un seul dépôt.

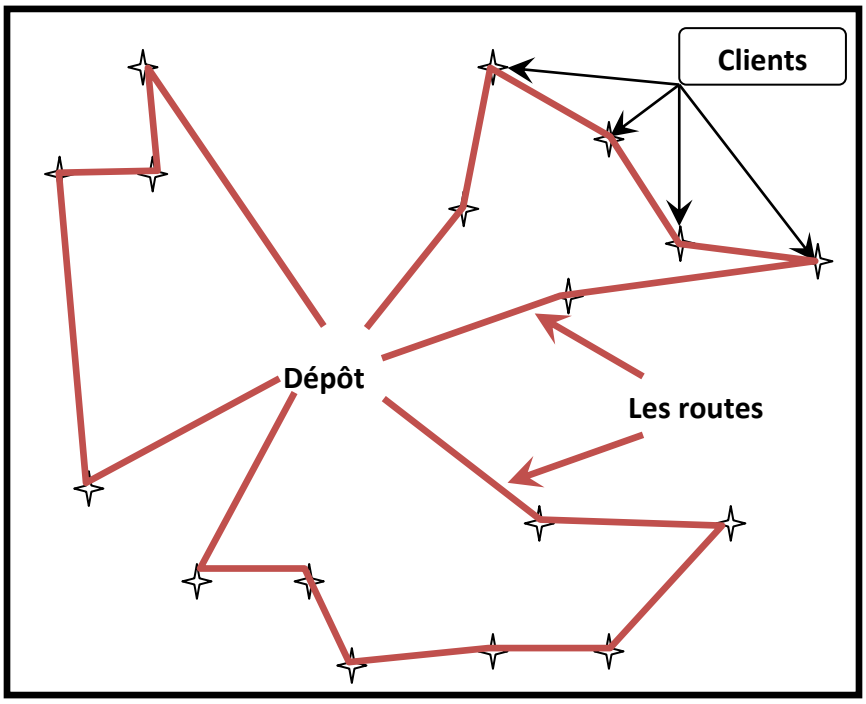


Figure 8 : une solution possible avec deux véhicules pour le problème de la Figure 7.

Le problème du voyageur de commerce, TSP, est un cas particulier du VRP qui a contribué à l'avancement de la recherche opérationnelle (Magnanti, 1981; Laporte, 1992a; Gendreau et al, 1992; Lawler et al, 1985; Renaud et al, 1996a, b et 2000…). Ce problème consiste à déterminer le tour de coût minimum qui, partant et retournant au dépôt, visite une et une seule fois chaque client.

Une formulation mathématique du TSP de base peut être exprimée comme ci-dessous. C'est la première formulation qui a été proposée par Dantzig, Fulkerson et Johnson (1954) :

$$Min \quad \sum_{i \neq j} c_{ij} x_{ij} \tag{1}$$

Sous les contraintes :

$$\sum_{j=0}^{n} x_{ij} = 1 \quad \forall i = 0,1,...,n \tag{2}$$

$$\sum_{i=0}^{n} x_{ij} = 1 \quad \forall j = 0,1,...,n \tag{3}$$

$$\sum_{i,j \in S}^{n} x_{ij} \leq |S| - 1 \quad \left(S \subset \{0,1,...,n\}, 2 \leq |S| \leq n-1\right) \tag{4}$$

$$x_{ij} \in \{0,1\} \quad \left(i, j = 0,1,...,n, \quad i \neq j\right) \tag{5}$$

Dans cette formulation, n est le nombre de clients, l'indice 0 est réservé pour le dépôt et les x_{ij} sont les variables de décision

$$x_{ij} = \begin{cases} 1 & \text{si le voyageur de commerce passe du point i au point j} \\ 0 & \text{sinon} \end{cases}$$

L'équation (1) représente la fonction objectif à minimiser, c'est le coût total de la tournée du voyageur de commerce. Les équations (2) et (3), appelées contraintes de degré ou de conservation de flux, stipulent que chaque client doit être visité une et une seule fois. La contrainte (4) est la contrainte d'élimination des sous-tours, elle sert à interdire des tournées qui ne passent pas par le dépôt.

Pour le problème de tournées de véhicules, VRP, on peut obtenir une formulation mathématique en se basant sur celle du TSP et en remplaçant la contrainte (4) par une contrainte qui impose que chaque tournée contienne un seul dépôt et en ajoutant une contrainte sur la capacité des véhicules.

2.2.3 Les différentes variantes du VRP

Le problème du VRP défini ci-dessus n'est qu'une formulation de base. Diverses contraintes peuvent se greffer à celui-ci, venant ainsi complexifier sa résolution. Ces restrictions peuvent survenir au niveau des routes (symétrique ou asymétrique), au niveau des caractéristiques des clients devant être servis (demandes, requis, etc.), au niveau des propriétés de la flotte de drones (autonomie, coûts associés à l'utilisation, endroits des dépôts, etc.) et finalement au niveau du déroulement des missions (contraintes au niveau du temps, périodes de pause entre les vols, etc.). Dans cette section, nous énumérerons ces éléments en nous basant sur les travaux de Crainic et Semet (2005), et Toth et Vigo (2002). L'intérêt de cet éventaire de contraintes est qu'il nous permettra de modéliser les paramètres (autonomie, moments opportuns pour la

visite des cibles…) de notre travail afin de refléter le plus possible une utilisation dans des conditions réelles de fonctionnement des UVAs.

VRP avec limite de capacité (Capacitated VRP ou CVRP)

Le CVRP est la version de base du VRP et qui prend en compte les capacités de chacun des véhicules dans l'analyse du problème. Dans le cas des UAVs, les drones sont limités entre autres par leur autonomie de vol. Puisque cette autonomie est limitée, les drones doivent revenir périodiquement à l'entrepôt pour faire le plein. Cette situation limite donc le nombre de clients qu'un drone peut visiter pendant chaque tour. Ainsi, la solution d'un CVRP est une série de tours afin que chaque client soit visité une fois et où chaque tour est limité par l'autonomie de l'UAV. Pour les CVRP, la demande est déterministe, il n'y a qu'un seul entrepôt, les véhicules sont identiques et seulement les capacités sont imposées. L'objectif est de minimiser les coûts totaux à servir toutes les cibles.

VRP avec une fenêtre de temps (VRP with Time Window ou VRPTW)

Le VRPTW est une extension du CVRP dans lequel des contraintes de temps s'ajoutent à la contrainte de capacité du véhicule. Pour les UAVs, ces contraintes de temps peuvent être des préséances pour certaines cibles (un drone doit visiter les cibles dans un ordre prédéterminé) ou imposer un intervalle de temps dans lequel un client doit être visité. Ainsi, chaque client se voit associer une fenêtre de temps que doivent respecter les UAVs. Le drone doit donc visiter chaque cible à l'intérieur de cette fenêtre et doit servir cette cible selon une période de temps déterminée. De plus, si le véhicule arrive en

avance, il doit attendre au moins jusqu'au moment du début de la période de temps prédéterminée avant de passer à la prochaine cible. Ces fenêtres peuvent être fixes ou souples.

VRP avec retours (VRP with Backhauls ou VRPB)

Le VRPB est une extension du CVRB dans lequel les véhicules peuvent à la fois faire des livraisons et des cueillettes et ce, dans le but réduire les retours à l'entrepôt. Le VRPB définit deux types de clients : les clients auprès desquels de la marchandise doit être livrée (« linehaul customers ») et des clients auprès desquels de la marchandise doit être récupérée (« backhaul customers »). L'une des contraintes des VRPB est que les clients qui doivent recevoir de la marchandise soient servis avant que les clients qui retournent de la marchandise commencent à être servis.

VRP avec ramassage et livraison (VRP with Pickup and Deliver ou VRPPD)

Le VRPPD est une autre généralisation du CVRP où chaque client nécessite une livraison et une cueillette. Ainsi, chaque client se voit attribuer deux quantités, une qui représente la quantité de marchandise devant être livrée et l'autre représentant la quantité de marchandise devant être ramassée.

VRP avec livraison divisée (Split Delivery VRP ou SDVRP)

Le SDVRP survient lorsqu'une flotte de véhicules homogène doit servir des clients (Archetti et al., 2006). Chaque client peut ainsi être servi par différents véhicules et peut être visité plus d'une fois. De plus, la demande des clients peut être plus grande que la capacité des véhicules.

VRP avec multiple entrepôt (Multi-Depot VRP ou MDVRP)

Les variantes des VRP décrites ci-dessus ont toutes en commun un entrepôt unique. Cependant, des situations peuvent survenir où une série de clients puisse être servie par plus d'un entrepôt. Les MDVRP décrivent ce type de VRP. Dans les MDVRP, chaque route commence et fini au même entrepôt, chaque client est visité qu'une seule fois par un véhicule et la demande totale de chaque route ne dépasse par la capacité du véhicule (Renaud et al., 1996).

Capacitated arc routing problems (CARP)

Le CARP est la contrepartie des VRPs. Alors que les VRPs utilisent les nœuds d'un graphe pour décrire les contraintes du service et de ressources, les CARPs utilisent les arcs. Les variantes des CARPs et les méthodes de résolution de ces problèmes sont similaires à celles des VRPs.

Finalement, les objectifs à atteindre pour chacune de ces variantes peuvent être différents selon les cas. L'objectif est de trouver les routes qui permettent de servir tous les clients et 1) minimiser les coûts totaux de transport, 2) minimiser la taille de la flotte utilisée, 3) minimiser le temps de transport ou 4) aspirer à une répartition équitable du travail ou même une combinaison de ces facteurs.

2.2.4 Les techniques de résolution

Le VRP est un problème bien connu de la programmation en nombres entiers qui entre dans la catégorie des problèmes NP-difficiles, signifiant que l'effort informatique exigé pour résoudre ce problème augmente exponentiellement avec la taille de celui-ci. Ainsi,

les méthodes exactes, qui permettent l'obtention d'une solution optimale, deviennent rapidement inutilisables pour des problèmes de grande taille en raison des temps de calcul élevés requis pour résoudre les VRP. La recherche s'est donc tournée vers l'utilisation de méthodes qui faisaient un compromis entre recherche de la solution optimale versus les temps de calcul, il est souvent souhaitable d'obtenir des solutions approximatives, qui peuvent être trouvées assez rapidement et sont suffisamment précises pour le but recherché. Le développement et l'application d'heuristiques ont donc été faits pour résoudre ce type de problème et plus tard, les métaheuristiques ont été appliquées afin d'améliorer les résultats des heuristiques tout en maintenant les temps de calcul à des niveaux respectables.

Les principales techniques utilisées pour résoudre le VRP sont les suivantes :

Les approches exactes

Ces approches tentent de trouver la solution optimale, ce qui demande beaucoup de ressource et de temps de calcul. Plusieurs méthodes exactes ont été utilisées dans la littérature pour résoudre les VRP. Toth et Vigo (2002) énumèrent plusieurs variantes des méthodes exactes, les deux plus utilisées étant la méthode par séparation et évaluation (Branch and Bound) avec éventuellement l'ajout de plans sécants (Branch and Cut).

La méthode de branch and bound (procédure par évaluation et séparation progressive) (Carinic et Semet, 2005; Fisher, 1994; Toth et Vigo, 2002) est une méthode générale dont l'idée est de diviser le problème initial en sous-problèmes plus petits structurés sous forme d'arbre, et de couper des branches de cet arbre (afin de ne pas avoir besoin de les

évaluer) grâce à un argument de majoration. Le problème de départ est un problème en nombres entiers ou mixte. L'arbre est alors un arbre binaire dont le nœud racine représente le problème initial relaxé (on considère toutes les variables continues). Les autres nœuds, chacun représente un problème de programmation linéaire (continu) qui possède le même objectif que le problème initial, les mêmes contraintes, plus des bornes sur les valeurs de certaines variables.

La méthode branch and cut (procédure par séparation et évaluation et des plans sécants) (Naddef et Rinaldi, 2002), part d'une solution dont plusieurs contraintes ont été relâchées. Ensuite, l'algorithme fait plusieurs itérations en réintroduisant les contraintes relâchées de façon progressive jusqu'à l'obtention de la meilleure solution. C'est une méthode qui combine la méthode branch and bound et la méthode des plans sécants. Cette dernière est une méthode plus ancienne. Son idée de base consiste à résoudre la relaxation continue du problème et, si le résultat x viole les contraintes d'intégralité, de générer une coupe, une contrainte, qui sépare x et l'ensemble des solutions admissibles. Une coupe est un hyperplan H défini par une équation linéaire.

Cependant, toutes ces méthodes exactes ne peuvent résoudre que des cas relativement petits (Cordeau et al., 2007) et leurs temps de calcul sont très grands et sont variables. Jusqu'à présent, les heuristiques restent la seule approche la plus fiable pour résoudre des cas concrets. Contrairement aux algorithmes exacts, les heuristiques sont mieux adaptées à la résolution de variantes de VRP impliquant des contraintes supplémentaires telles que les fenêtres de temps (Cordeau et al., 2002), des collectes et des livraisons (Desaulniers et al., 2002), des visites périodiques (Cordeau et al., 1997).

Les heuristiques classiques

Les méthodes heuristiques ont été développées afin de trouver un compromis entre une solution acceptable, mais non optimale, et un temps de calcul raisonnable. Ces méthodes, qui permettent l'inclusion d'un plus grand nombre de contraintes, peuvent se diviser en trois catégories (Laporte et Semet, 2002) :

- Les heuristiques constructives construisent peu à peu une route en choisissant les nœuds à inclure de façon séquentielle, tout en gardant un œil sur le coût de la solution. On compare ainsi le coût de la solution obtenue avec celui de la solution initiale en recherchant des économies. La solution la plus économique sera celle retenue. Cependant, ce type de méthode ne contient pas une phase d'amélioration en soi. L'algorithme de Clark et Wright (1964) et l'amélioration multi-route (Thompson et Psaraftis, 1993; Van Breedam, 1994; Kinderwater et Savelsbergh, 1997) sont les méthodes heuristiques constructives les plus utilisées.
- Les heuristiques en deux phases décomposent le problème en ses deux composantes naturelles : le regroupement des sommets en routes possibles (partition) et la construction des routes réelles (parcours). Donc, les heuristiques en deux phases commencent en divisant le terrain à parcourir en regroupant des clients. Puis, l'algorithme construit des routes pour chacun de ces regroupements tout en respectant les contraintes du problème. Les algorithmes les plus utilisés sont : le « *cluster-first, route-second* » (Taillard, 1993; Fisher et Jaikumar, 1981);

le « *petal algorithm* » (Ryan, Hjorring et Glover, 1993); le « *sweep algorithm* » (Renaud et Boctor, 2002).

- Les méthodes d'amélioration tentent d'améliorer une solution en effectuant des échanges d'un segment de route contenant certains clients avec une autre route. L'action est répétée itérativement jusqu'à l'obtention d'une solution acceptable.

Les métaheuristiques

Les métaheuristiques sont des stratégies qui permettent de guider la recherche d'une solution optimale. Le but visé par les métaheuristiques est d'explorer l'espace de recherche efficacement afin de déterminer des solutions quasi optimales. Elles forment un ensemble de méthodes utilisées en recherche opérationnelle pour résoudre des problèmes d'optimisation réputés difficiles, généralement combinatoires. Elles permettent, dans des temps de calcul raisonnables, de trouver des solutions, peut-être pas toujours optimales, mais très proches de l'optimum; elles se distinguent en cela des méthodes exactes, qui garantissent certes la résolution d'un problème, mais au prix d'un temps de calcul énorme qui les rendent inapplicables pour des problèmes pratiques.

Les métaheuristiques mettent de l'avant des algorithmes qui explorent l'ensemble des solutions possibles afin de trouver de bonnes solutions. Cette exploration peut se faire de deux façons : soit en faisant évoluer une population (un ensemble de solutions), soit en commençant avec une seule solution qui est souvent déterminée par une heuristique, à laquelle on applique un algorithme afin de l'améliorer (les solutions manipulées dans les deux cas peuvent être admissibles ou non). Ceci permet de classer les métaheuristiques

en deux familles; celles qui travaillent avec une population de solutions et celles qui ne manipulent qu'une seule solution à la fois. Les méthodes qui tentent itérativement d'améliorer une solution sont appelées méthodes de recherche locale. La recherche Tabou et le Recuit Simulé sont des exemples de ces méthodes de recherche locale. Les méthodes qui travaillent sur une population de solutions sont des méthodes qui vont, grâce à des manipulations, créer des générations de solutions : à partir d'une population de solutions ou de toutes les générations précédentes, ces méthodes créent une nouvelle population qui présente généralement de meilleures caractéristiques que les précédentes. L'exemple le plus connu de ces méthodes est l'algorithme génétique.

Les métaheuristiques permettent des solutions intermédiaires non admissibles et elles possèdent une capacité de mémoire afin d'enregistrer les solutions utilisées, mais non retenues. Différentes méthodes heuristiques ont été développées, spécialement pour les cas du VRP; quelques-unes sont élaborées ci-dessous (Gendreau et al., 2002) :

- La recherche Tabou a été proposée pour la première fois par Glover (1986) et est rapidement devenue l'une des meilleures des méthodes de recherche locale et la plus répandue pour l'optimisation combinatoire. La méthode effectue une exploration de l'espace de solutions en passant d'une solution donnée, à l'itération t, à la meilleure solution dans un voisinage de celle-ci. Pour éviter les extrema locaux, la méthode utilise un mécanisme tabou pour empêcher le cyclage. Une manière naïve de prévenir les cycles est d'interdire le processus de retourner à des solutions rencontrées auparavant, mais cela demanderait beaucoup de mémoire. Au lieu de cela, on interdit seulement certains attributs des solutions

passées pour un certain nombre d'itérations. Ce mécanisme est souvent appelé la mémoire à court terme. D'autres caractéristiques de la recherche Tabou sont la diversification et l'intensification. L'objectif de la diversification est de s'assurer que le processus de recherche ne se limitera pas à une partie de l'espace solution. Pour cela, la diversification conserve en mémoire certaines caractéristiques des solutions visitées et pénalise les mouvements qui se répètent fréquemment. Ceci est souvent appelé la mémoire à long terme. L'intensification consiste à effectuer une recherche accentuée autour de la meilleure des solutions connues. Le voisinage dans la recherche Tabou est en général défini par l'échange de plusieurs clients entre les routes (Osman, 1993). Plusieurs articles et livres ont été écrits sur la recherche Tabou parmi eux Glover et Laguna (1997), et Hertz et de Werra (1991). Et parmi les algorithmes de recherche Tabou les plus connus, on cite : le Taburoute de Gendreau, Hertz et Laporte (1994); l'algorithme de Taillard (1993); la procédure de mémoire adaptative de Rochat et Taillard (1995); l'algorithme de Xu and Kelly (1996).

- Le recuit simulé (« simulating annealing ») utilise une solution de départ à laquelle une énergie E_0 est donnée et qui dépend des critères que l'on cherche à optimiser. Par la suite, des itérations sont effectuées afin d'évaluer différentes solutions. Si la nouvelle solution a une énergie plus basse que la solution initiale, elle est mise de côté et une autre solution est évaluée. Si l'énergie de la solution est plus grande, celle-ci peut remplacer la solution initiale selon une probabilité qui dépend de la différence d'énergie entre les deux solutions.

- Les méthodes avec populations : le mécanisme fondamental des méthodes avec population est la combinaison de plusieurs solutions pour créer de nouvelles solutions de meilleure qualité. Parmi ces méthodes, on trouve les algorithmes génétiques (Reeves, 2003), et les algorithmes mimétiques (Moscato et Cotta, 2003). Les algorithmes génétiques opèrent sur une population de solutions codées appelées chromosomes. À chaque itération (génération) les opérations suivantes sont appliquées k fois : sélectionner deux chromosomes parent; générer deux descendants de ces parents en utilisant un opérateur de croisement; appliquer une mutation aléatoire à chaque descendant avec une probabilité donnée, retirez les 2k pires éléments de la population et les remplacer par les 2k descendants qu'on vient de générer. Plusieurs méthodes de croisements ont été proposées (Drezner, 2003). La procédure de mémoire adaptative présentée par Rochat et Taillard (1995) pour la résolution du VRP est basée aussi sur l'idée de combiner des solutions pour en générer d'autres. Rochat et Taillard (1995) ont construit des routes à partir de plusieurs bonnes solutions et ils les ont utilisés comme base pour la construction de la nouvelle génération.
- La colonie de fourmis (Gambardella et al. 1999) est un algorithme inspiré de fourmis recherchant de la nourriture. Les fourmis utilisent un système de communication basé sur la phéromone : le chemin parcouru par une fourmi à partir de la colonie jusqu'à une source de nourriture est marqué pour que les autres membres de la colonie puissent suivre le chemin. Les meilleures sources de nourriture, celles qui sont abondantes et situées près de la colonie, attirent le

plus de fourmis. Ainsi, les chemins partant de la colonie vers ces sources de nourriture (les meilleures solutions) sont ceux qui auront la plus grande concentration de phéromone. Ce principe peut être appliqué afin de rechercher la meilleure solution à un VRP. Des fourmis artificielles sont lancées dans l'espace de solutions d'un problème. Des valeurs, l'équivalent de la route de phéromone, sont attribuées à ces solutions en fonction de certains critères de sélections. La solution ayant la plus grande valeur est celle qui est retenue.

2.3 Conclusion

Suite à ce survol de la revue de la littérature sur la modélisation, on peut affirmer qu'il reste encore beaucoup de travail à faire. Puisque la plupart des approches sont issues de gens spécialisés dans le domaine des « contrôles de systèmes», les travaux sont donc orientés côté technique et loin des problèmes opérationnels et stratégiques. Quand on s'intéresse à la partie technique on se contente en général des approches très simples et n'intégrant pas les contraintes réelles.

Dans tous les travaux, le principe de base est la modélisation du terrain sous forme d'un graphe suivi de l'application d'un algorithme ou heuristique de calcul de plus court chemin. Cependant aucun travail ne couvre le problème dans sa totalité et avec toutes les contraintes.

Aussi, il reste encore beaucoup de travail à faire au niveau de la modélisation du problème de base en plus d'intégrer à la modélisation du terrain, les contraintes d'obstacles, de danger, la capacité de la flotte, sa cinématique et son autonomie. De plus,

il faut trouver des approches algorithmiques qui permettent de trouver rapidement des solutions de qualité raisonnable.

2.4 Notre démarche

Le principe de base derrière notre démarche est l'intégration de plusieurs outils et disciplines pour aboutir à identifier une solution pratique à une problématique réelle. La Figure 9 résume notre démarche : on voit en haut de la figure le terrain à surveiller avec une flotte d'UAVs et on doit planifier les routes que doit prendre chaque UAV pour assurer la réalisation de la mission dans les meilleures conditions. Deux solutions sous forme de trajectoires réelles (tenant compte de toutes les contraintes de contournement d'obstacles, de danger et de cinématiques de l'avion) sont présentées en bas de la figure. Une solution où on tolère le risque et l'autre où on ne tolère aucun risque.

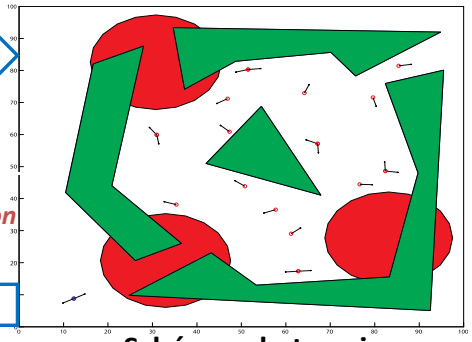
La démarche s'articule comme suit : premièrement une schématisation du terrain est faite par des polygones, qui représentent les obstacles, et par des cercles, qui représentent la portée des radars. On y indique la position des cibles connues d'avance, il s'agit dans notre cas d'une planification de missions de surveillance. Ensuite le terrain est modélisé sous forme d'un graphe orienté. Ceci est fait intelligemment et non par un quadrillage (Pelosi, 2010) pour garantir la même précision avec moins de calcul. Sur ce graphe un problème de plus court chemin avec contrainte de ressources est résolu pour évaluer les différents chemins possibles. Nous avons plusieurs chemins car entre chaque paire de cibles et le point de départ, on cherche des chemins courts avec différents seuils de risque (ceci est nécessaire pour le calcul des routes avec un risque limite, voir la

section sur la planification de la mission). Ainsi nous obtenons un multi-graphe sur lequel un Tabou à double liste est appliqué pour obtenir les meilleures routes, c.-à-d. celle minimisant le coût de la mission et respectant les contraintes cinématiques de l'avion, et les contraintes de risque et d'autonomie.

Le terrain

(Carte géographique du terrain et les coordonnées des cibles et des radars)

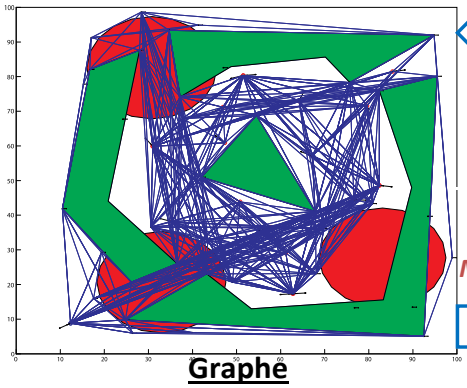
Synthèse du terrain



Schémas du terrain

(Carte plus simple contenant : enveloppe des obstacles et des radars et les positions des cibles et du point de départ.)

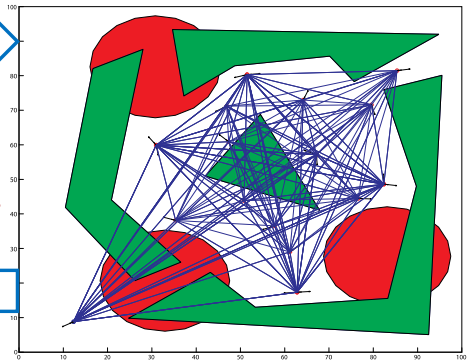
Modélisation du terrain



Graphe

(Graphe modélisant le terrain : les nœuds sont des points particuliers, voir modélisation du terrain, et les arcs sont des arcs à double valence.)

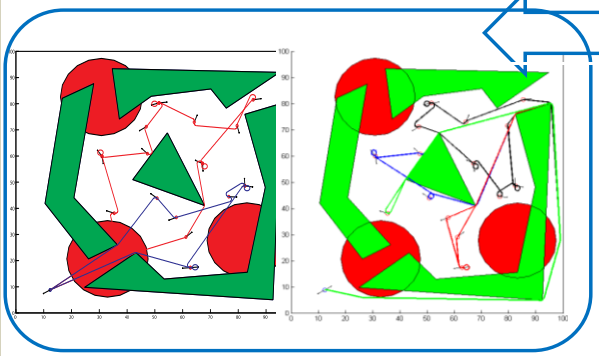
Calcul du Multi graphe



Multi Graphe

(Graphe servant au calcul des routes – trajectoires. Les arcs ne sont que schématiques chaque trait représentant plusieurs arcs – trajectoires réelles ne passant pas par les obstacles.)

Calcul des routes



Solution finale

(Ici on a la représentation de deux solutions réelles : une qui tolère plus de risque et l'autre qui ne tolère aucun

Figure 9 : Notre démarche

CHAPITRE 3

PLUS COURT CHEMIN CONTINU NON HOLONOME DANS UN ENVIRONNEMENT À RISQUE ET EN PRÉSENCE D'OBSTACLES

Le problème du plus court chemin continu en présence d'obstacles, de danger et avec des contraintes de cinématique – c'est-à-dire non holonome – est une des questions clés dans la planification des trajectoires des UAVs. Ce problème est reconnu comme NP-difficile (Chandler et al., 2002; Schumacher et al., 2002; Bellingham et al., 2001). La majorité des travaux traitent une partie des contraintes de ce problème en passant par un graphe, dont les nœuds sont des points intermédiaires, sur le lequel on applique un algorithme de plus court chemin, par exemple, celui de Dijkstra (Kim et Hespanha, 2003; Kuwata, 2003; De Mot et al., 2002). Cependant, il n'existe aucune solution tenant compte de toutes ces contraintes à la fois et surtout de la contrainte de non-holonomie (contrainte de cinématique du véhicule).

L'objectif derrière ce chapitre est de trouver le chemin le plus court et le moins risqué ou dont le risque encouru ne dépasse pas un seuil fixé. A priori, c'est une problématique multiobjectif. Nous avons deux objectifs : la longueur du chemin à minimiser et le risque encouru par le véhicule à réduire. Cependant dans notre cas, il s'agit d'une situation où

l'armée fixe la part de risque à assumer. Elle est fixée via un paramètre alpha qui apparait dans la fonction objectif globale et qui nuance l'importance du risque par rapport au coût de la mission ou sous forme d'un risque limite que la trajectoire ne peut pas excéder.

Dans ce chapitre, nous présentons une approche de résolution qui tient compte de toutes les contraintes, celles présentées dans le chapitre décrivant la problématique, et qui optimise l'objectif fixé par l'armée (qui peut être la minimisation d'une combinaison fixée de la longueur de la trajectoire et du risque sur cette dernière ou la minimisation de la longueur de la trajectoire tout en gardant le risque en deçà d'un certain seuil donné). Notre approche est constituée d'une modélisation du terrain en un graphe orienté intégrant toutes les contraintes et de deux algorithmes de calcul de plus court chemin sur un graphe. Le premier est une adaptation d'algorithme de Dijkstra (1959) classique et le deuxième est une adaptation des travaux de Beasley et Christofides (1989) sur le calcul d'un plus court chemin avec des contraintes de ressources. Notre approche est novatrice car c'est la première fois qu'on traite ce problème dans sa totalité. L'essentiel de la contribution est dans la modélisation du terrain en un graphe orienté généré intelligemment pour n'y garder que les éléments essentiels et ainsi rendre le calcul plus rapide. Cette génération est faite de sorte à permettre d'intégrer, au fur et à mesure, de nouvelles données sur le terrain, ce qui facilite la tâche d'exécution de la mission, De plus, cette approche peut être appliquée à d'autres types de robots non holonomes comme les robots sous-marins et les robots sur roues.

Le chapitre présent s'articule comme suit : la section 1 décrit la problématique du plus court chemin non holonome avec des contraintes d'obstacles et de dangers. La problématique s'inscrit dans le projet de la thèse qui concerne la planification, le design et le contrôle de trajectoires d'une flotte d'UAVs en vue d'une mission de surveillance d'un terrain hostile. La section 2 présente deux formulations mathématiques du problème. La section 3 présente l'approche de modélisation du terrain et les algorithmes de calcul du plus court chemin. Les résultats obtenus par des tests sur différentes configurations de terrain sont donnés dans la section 4. La section 5 conclut ce chapitre et en présente les perspectives.

3.1 Le problème du plus court chemin non holonome

Dans ce chapitre, nous allons développer un outil qui permet de trouver des trajectoires qui respectent toutes les contraintes et qui soient les plus courtes possibles. Avant d’énoncer le problème, on va revenir sur le terrain, les UAVs et leurs caractéristiques.

3.1.1 Le terrain, les obstacles et les dangers

La mission se déroule dans un terrain donné, qui se situe dans une zone ennemie et dans des conditions météorologiques données. Cela donne lieu à la présence d'obstacles — les reliefs et les zones impossibles de navigation – et de zones de danger – les zones sous surveillance par les radars de l'ennemi. La Figure 10 montre un exemple de terrain. Dans cette figure, les polygones représentent les obstacles et les cercles représentent les zones de danger. La configuration de départ et la cible sont représentées par des points, dont

les coordonnées sont la position du véhicule, et des vecteurs qui indiquent l'orientation du véhicule en ces points.

Les vols se font à une altitude constante, et l'orientation du drone est importante pour savoir si la cible est bien visitée. Ainsi nous sommes amenés à définir la notion de configuration et de l'espace E de configurations. Une configuration possible de l'avion est une position de coordonnées x et y et un angle θ donnant l'orientation du véhicule. E, l'espace des configurations, est l'ensemble de toutes les configurations possibles dans le terrain visé par la mission. Ainsi l'ensemble des cibles à visiter est un ensemble de configurations, sous-ensemble de E.

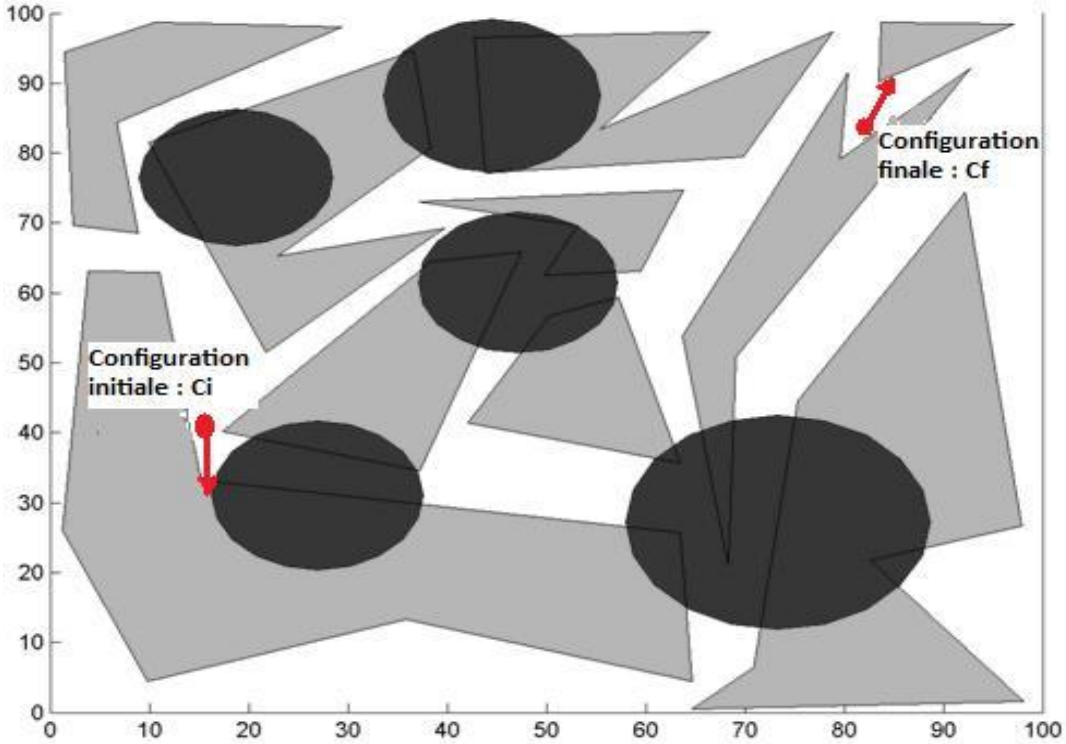


Figure 10 : Exemple de terrain (les polygones représentent les obstacles et les cercles les zones de danger).

Les obstacles sont des reliefs ou des zones non navigables. Ils sont définis par leur contour. Ce contour peut être approximé par une ligne polygonale en utilisant des algorithmes d'approximation de courbes fermées comme celui présenté par Kumar et al. (2004). Ainsi les obstacles seront considérés comme des polygones (convexes ou non) comme sur la Figure 10. L'ensemble des points situés dans les obstacles sera noté $\mathcal{O}$.

Les zones de danger sont les zones sous contrôle de l'ennemi. Elles sont surveillées par des radars dont la portée est définie par un rayon donné R. Ces zones seront représentées par des disques de rayon R. Le drone est hors danger quand il est en dehors de ces disques. Sinon il est dans une zone de danger où la navigation est permise mais risquée car il y a une probabilité non nulle que l'ennemi détecte le drone. Ce risque est lié à la distance entre l'avion et les radars, les centres des zones de danger.

3.1.2 L'UAV et les systèmes non holonomes

Il existe deux types de systèmes en robotique (Bellaïche et al., 1998; Chitour et al., 2009) : les systèmes holonomes, qui n'ont pas de restriction sur le déplacement, tout chemin dans l'espace des configurations est un mouvement autorisé; les systèmes non holonomes, qui sont soumis à des contraintes cinématiques, les seuls mouvements autorisés sont des chemins dans l'espace de configurations tels que la dérivée de la position vérifie certaines contraintes incluant :

- contraintes d'angle maximum de braquage : les drones (UAV); robots mobiles à roues;

- contraintes de roulement sans glissement : robots mobiles à roues, manipulation par des mains robotisées;
- conservation du moment angulaire : robots manipulateurs flottants dans l'espace; satellites avec roues à inertie; robots astronautes, plongeurs, sauteurs (phase de vol);
- autres sources de contraintes non holonomes : loi de Lorentz-Wong en électromagnétisme; réduction par symétrie de cinématiques holonomes.

Le drone, qui nous intéresse ici est dans la réalité un robot non holonome, il ne peut suivre n'importe quel chemin dans l'espace de configurations. Sa trajectoire doit respecter la contrainte cinématique de rayon minimum de giration qui est équivalente à la contrainte d'angle maximum de braquage :

$$\begin{cases} \dot{x} = v\,cos(\theta) \\ \dot{y} = v\,sin(\theta) \end{cases} \\ -\phi_{max} \leq \dot{\theta}(t) \leq \phi_{max} \qquad (0)$$

Dans l'équation (0), (x, y) représente les coordonnées du drone, θ représente sa direction et $\dot{x}, \dot{y}, \dot{\theta}$ sont respectivement les dérivées de x, y, θ. Lors du vol de l'UAV, sa trajectoire est définie par une suite de commandes : la vitesse linéaire à tout instant t, $v(t)$ et l'angle de braquage à tout instant t, $\phi(t) = \dot{\theta}(t)$.

3.1.3 Le problème

Une mission de surveillance dans un terrain réel contrôlé par l'ennemi est, donc, la visite de plusieurs cibles, configurations, sans rencontre d'obstacles et en évitant autant que

possible les zones de danger, les radars. C’est-à-dire en minimisant le risque ou en le gardant en dessous d’un certain seuil. La planification de cette mission nécessite de savoir à tout moment le plus court chemin réalisable entre deux configurations données.

Notre problème est, alors, de déterminer la trajectoire la plus courte et la moins risquée (ou avec un risque limite) entre ces deux configurations en respectant les contraintes cinématiques de l'avion, évitant les obstacles et minimisant le temps passé dans les zones de danger, voire les éviter si cela est possible.

Les hypothèses considérées sont :

- le problème est en trois dimensions, mais vu que les vols se font à une altitude constante, on va modéliser le terrain en deux dimensions uniquement;
- une configuration du véhicule est définie par les coordonnées (x, y) de l'avion et l’orientation de ce dernier qui sera modélisé par un angle θ, voir Figure 10;
- les obstacles sont définis par des polygones convexes ou non convexes et on suppose que les avions peuvent longer ces derniers;
- les zones de danger sont définies par des disques dont le centre et le rayon sont donnés. On considère que le radar a une portée limitée définie par ce rayon;

- la non-holonomie de la trajectoire pour notre cas, celui des avions, est définie par la contrainte cinématique, ou de manœuvrabilité de l'avion reliée à son angle limite de braquage (équation (0)). Dans la suite, elle sera caractérisée par la donnée de cet angle maximum ou par la donnée d'un rayon de courbure minimum;
- il s'agit dans notre cas d'avion et il n'y a donc pas de marche arrière permise.

Le problème est donc de **déterminer le Plus Court Chemin Continu Non holonome dans un terrain avec des Obstacles et des zones de Risque** (PCCC_NH_OR).

3.2 Modélisation et Formulations mathématiques

Nous considérons le problème introduit ci-dessus. Nous cherchons le plus court chemin non holonome entre deux configurations, celle de départ de l'avion et celle d'arrivée, dans un milieu jonché d'obstacles et sous surveillance de radars. Pour tenir compte du risque nous définissons deux formulations du problème, une avec le risque introduit dans la fonction objectif et une avec le risque sous forme d'une contrainte avec un risque limite. Les deux formulations sont présentées ci-dessous après l'introduction de quelques notations.

3.2.1 Notations

Dans ces formulations mathématiques et dans la suite de cet article, nous utiliserons les notations suivantes :

- E : l'espace des configurations (x, y, θ);
- $\mathcal{O}$: l'ensemble des obstacles. C'est l'ensemble des points (x, y) du terrain appartenant aux obstacles;
- α : le poids associé au risque de la trajectoire. Dans notre cas, ce paramètre est une donnée fixée par la défense canadienne selon le type de la mission, $\alpha \in [0, +\infty]$;
- t : indique le temps qui s'écoule entre les instants 0, le moment de départ de l'avion, et t_F, l'instant d'arrivée à la cible;
- $(x(t), y(t), \theta(t))$: le triplet indiquant la configuration de l'avion à l'instant t. $(x(t), y(t))$ représente la position de l'avion sur le terrain et $\theta(t)$ donne l'orientation de ce dernier;
- $r\big(x(t), y(t)\big)$: la densité du risque associé au point $(x(t), y(t))$. Le risque est dû, dans notre cas, à la présence de radars.
- ϕ_{max} : l'angle de braquage maximum que peut réaliser l'avion. Il représente la contrainte de cinématique associée au véhicule. À tout instant t, l'angle de braquage est : $\phi = \dot{\theta}(t)$, qui est la dérivée de θ;

- $v(t)$: la vitesse de l'avion à l'instant t.

En se basant sur cette notation, la distance parcourue par le drone entre deux configurations définies aux instants t_1 et t_2 peut-être formulée via le calcul de l'intégrale suivante : $\int_{t_1}^{t_2} \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2}\, dt$. Et le risque encouru par ce drone est obtenu par l'intégration de la fonction densité de risque :$\int_{t_1}^{t_2} r\big(x(t), y(t)\big)\, dt$.

Les deux formulations ci-après sont basées sur le travail de Dubins (1957) et sur la représentation des contraintes de cinématiques d'un drone (Bellaïche et al., 1998; Chitour et al., 2009).

3.2.2 Les variables de décision

La solution de notre problème est une suite de commandes $v(t)$ et $\phi(t)$ qui dictent au drone la vitesse et l'angle de braquage à appliquer à tout instant t. ϕ et v sont les variables de décisions.

3.2.3 Formulation 1 : le risque dans la fonction objectif

Étant donné une configuration initiale $(x(0), y(0), \theta(0))$ et une configuration finale $(x(t_F), y(t_F), \theta(t_F))$, le problème du plus court chemin non holonome entre ces deux configurations peut être formulé comme suit :

L'objectif :

$$Min\left[\left(\int_o^{t_F} \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2}\, dt\right) + \alpha\left(\int_o^{t_F} r\big(x(t), y(t)\big)\, dt\right)\right] \qquad (1)$$

Sous les contraintes :

$$\begin{cases} \big(x(0), y(0), \theta(0)\big) = point\ origine; \\ \big(x(t_F), y(t_F), \theta(t_F)\big) = point\ cible; \end{cases} \tag{2}$$

$$\begin{cases} \dot{x}(t) = v(t)cos\big(\theta(t)\big) \\ \dot{y}(t) = v(t)sin\big(\theta(t)\big) \end{cases} \forall\ t\ \in\ [0, t_F]; \tag{3}$$

$$-\phi_{max} \leq \dot{\theta}(t) \leq \phi_{max}\ \forall\ t\ \in\ [0, t_F]; \tag{4}$$

$$(x(t), y(t)) \notin \mathcal{O}\ \forall\ t\ \in\ [0, t_F]; \tag{5}$$

Dans cette formulation les variables de contrôle – de décision – sont $v(t)$ et $\phi = \dot{\theta}(t)$ que le programme devra déterminer. L'équation (1) représente la fonction objectif. Elle est formée de deux parties, la première intégrale calcule la longueur de la trajectoire et la deuxième intégrale calcule le risque sur la trajectoire. α est le poids associé au risque. Dans notre cas, ce paramètre est une donnée fixée par la défense canadienne selon le type de la mission. Les deux contraintes (2) assurent qu'on parte de la bonne configuration et qu'on atteigne la bonne cible avec la bonne orientation. (3) sont des équations différentielles qui relient les commandes de contrôle à la position de l'avion. (4) est la contrainte cinématique reliée au rayon de courbure minimum représenté ici par $\dot{\theta}(t)$, la variation de la direction de l'avion qui doit être confinée entre $-\phi_{max}$ et ϕ_{max}, l'angle de braquage maximum. Et la dernière contrainte (5) signifie que tous les points de la trajectoire sont en dehors de la zone des obstacles.

3.2.4 Formulation 2 : le risque dans la contrainte

L'objectif :

$$Min\left[\left(\int_{o}^{t_F}\sqrt{\dot{x}(t)^2+\dot{y}(t)^2}\,dt\right)\right] \qquad (6)$$

Sous les contraintes :

$$\begin{cases}\big(x(0),y(0),\theta(0)\big)=point\ origine;\\ \big(x(t_F),y(t_F),\theta(t_F)\big)=point\ cible;\end{cases} \qquad (2)$$

$$\begin{cases}\dot{x}(t)=v(t)cos\big(\theta(t)\big)\\ \dot{y}(t)=v(t)sin\big(\theta(t)\big)\end{cases}\ \forall\ t\ \in\ [0,t_F]; \qquad (3)$$

$$-\phi_{max}\leq\dot{\theta}(t)\leq\phi_{max}\ \forall\ t\ \in\ [0,t_F]; \qquad (4)$$

$$(x(t),y(t))\notin\mathcal{O}\ \forall\ t\ \in\ [0,t_F]; \qquad (5)$$

$$\int_{o}^{t_F}r\big(x(t),y(t)\big)\,dt\ \leq r_{max}; \qquad (7)$$

Cette formulation est presque identique à la première hormis deux différences : la fonction objectif (6) qui ne contient qu'un seul terme celui du calcul de la longueur de la trajectoire, et la contrainte (7) qui définit une limite pour le risque sur la trajectoire.

Les deux formulations ci-dessus présentent les mêmes caractéristiques : la continuité du temps, la présence des équations et des inéquations différentielles, la contrainte (5) d'évitement des obstacles et la non-linéarité du problème. Une discrétisation du temps et de l'espace permettrait de reformuler le problème en un programme non linéaire mais avec un grand nombre de variables et de contraintes – cette formulation dépendra de la précision voulue au niveau des résultats.

Ceci pousse à trouver une autre alternative de résolution. Notre idée de base consiste à passer d'un problème continu en un problème discret. Passer de la recherche d'un Plus Court Chemin Continu Non holonome dans un terrain avec des Obstacles et des zones de Risque (PCCC_NH_OR) en la recherche d'un plus court chemin (PCC) sur un graphe orienté. Ceci consiste en deux phases : 1) modélisation du terrain avec toutes les contraintes en un graphe orienté; 2) puis calcul du plus court chemin sur ce graphe. Ce dernier calcul se fait de deux façons selon les deux formulations mathématiques présentées ci-dessus. La première en intégrant le risque dans l'objectif - ce calcul utilise un algorithme de Dijkstra classique - et la deuxième en considérant un risque limite - ce calcul utilise un algorithme de Dijkstra modifié. Ces deux algorithmes sont détaillés dans la section 3.3. Les figures, Figure 11 et Figure 12, illustrent notre approche : la Figure 11 est un terrain avec des obstacles, des zones de danger et les configurations initiale et finale. La Figure 12 est le graphe correspondant à ce terrain (les ronds représentent les sommets du graphe et les lignes les arcs). La sous-section suivante présente notre modélisation du terrain.

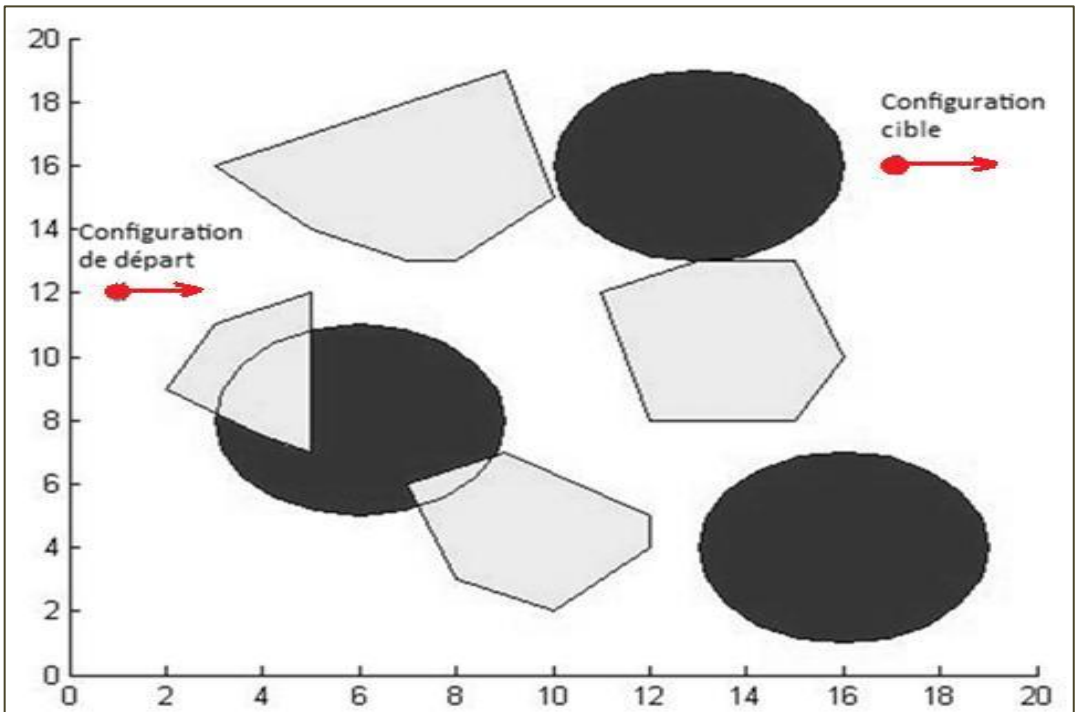


Figure 11 : Exemple de terrain avec des obstacles et des zones de danger.

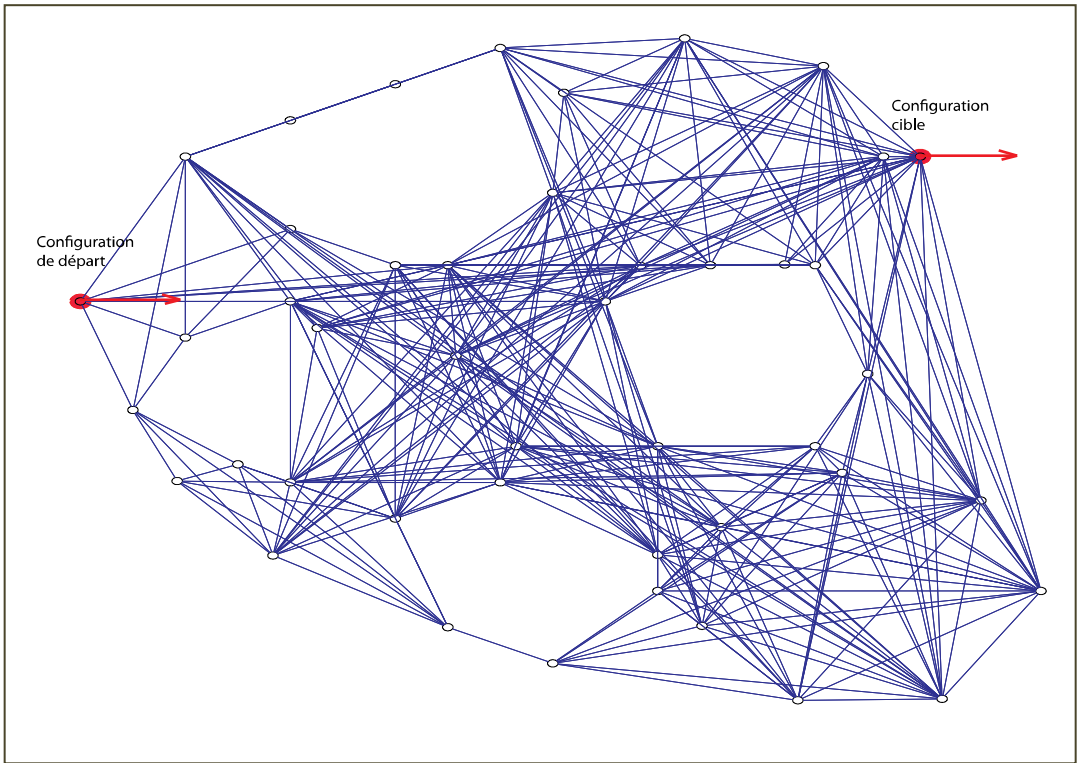


Figure 12 : Le graphe correspondant au terrain de la Figure 11 et sur lequel le plus court chemin sera calculé.

3.2.5 Modélisation du terrain

3.2.5.1 Idée de base

L'idée clé qui constitue l'essentiel du travail est la modélisation du terrain en un graphe orienté avec des arcs à double valence (la longueur de l'arc et le risque associé à l'arc). Cette modélisation inclut :

- la détermination des points intermédiaires : Transformation du problème continu en un problème discret, c'est-à-dire modélisation du terrain en un graphe orienté via un graphe de visibilité. Cette phase intègre la modélisation des obstacles;
- l'élagage des nœuds que l'avion ne pourra pas visiter;
- l'intégration des zones de danger dans le graphe;
- l'intégration de la non-holonomie (les contraintes cinématiques) via les courbes de Dubins et le calcul de la longueur des arcs;
- un élagage des arcs pour éliminer les chemins non réalisables;
- le calcul du risque.

Ci-après, nous présentons et illustrons le détail de modélisation et l'intégration de chacun des points ci-dessus et ce dans le même ordre que leur présentation.

3.2.5.2 Détermination des points intermédiaires : graphe de visibilité

Le drone part de la configuration initiale Ci et doit se rendre à la configuration finale Cf. Durant son vol, il passera par des points clés, se sont les points intermédiaires. A priori, ces derniers peuvent être n'importe quelle configuration du terrain n'appartenant pas aux obstacles.

En se basant sur ce constat, certains auteurs dont De Mot et al. (2002), Kim et Hespanha (2003) proposent une discrétisation du terrain. Sachant qu'une configuration de l'avion est définie par ses coordonnées x, y et par son orientation θ, ils proposent le découpage du terrain selon les deux axes et selon l'angle θ. Ceci pose beaucoup de problèmes : 1) un grand nombre de points intermédiaires, donc un graphe avec beaucoup de nœuds; 2) un dilemme entre la précision et la charge de calcul (une bonne précision implique une discrétisation plus fine, donc une charge de calcul plus grande); 3) L'introduction de points intermédiaires non utiles (voir Figure 13).

En effet une discrétisation d'un terrain en 10 x 10 avec la direction θ subdivisée en 10, donne, pour ce petit exemple, 1 000 nœuds, a priori 499 500 arcs et 2 977 000 trajectoires à évaluer (pour tenir compte de la non-holonomie). C'est un graphe énorme! De plus, une bonne partie de ces nœuds n'est pas à considérer. Sur la Figure 13, les points encerclés sont des points intermédiaires (des nœuds introduits par la discrétisation) qui ne sont pas utiles car aucune trajectoire optimale ne passera par ces points.

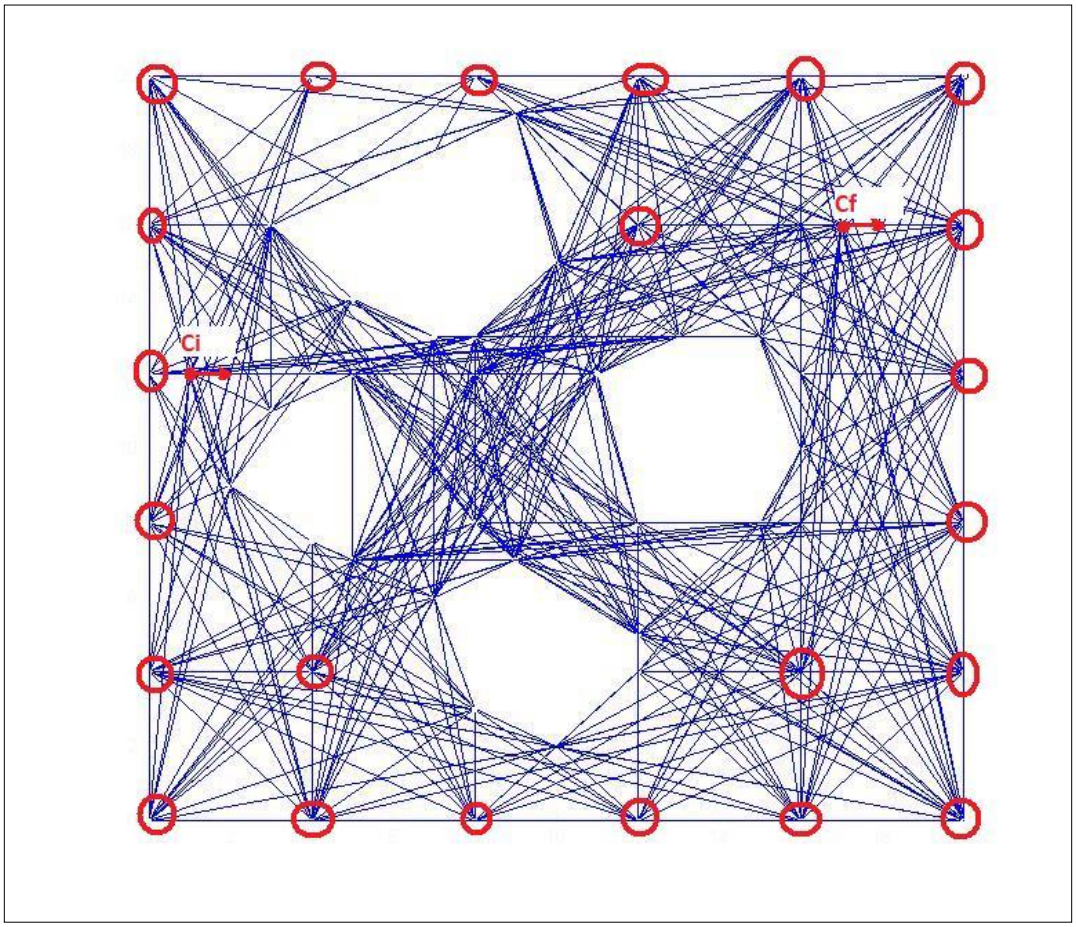


Figure 13 : Un exemple de graphe généré par maillage. Les points encerclés sont des exemples de points intermédiaires non utiles

Pour traiter cette problématique, nous n'allons introduire que les points intermédiaires potentiellement nécessaires. Ceci est possible en se basant sur le graphe de visibilité (Richards et al., 2002). C'est un graphe, voir Figure 6, qui est basée sur la notion de champ de vision, ou angle solide. Les seuls points à y considérer sont les points du plan dans le champ de vision du véhicule entre le point de départ et le point cible et qui représentent un changement dans le milieu. Ce sont, à priori les sommets des obstacles. Ces points de passage permettent de définir un graphe orienté $G = (N; A)$.

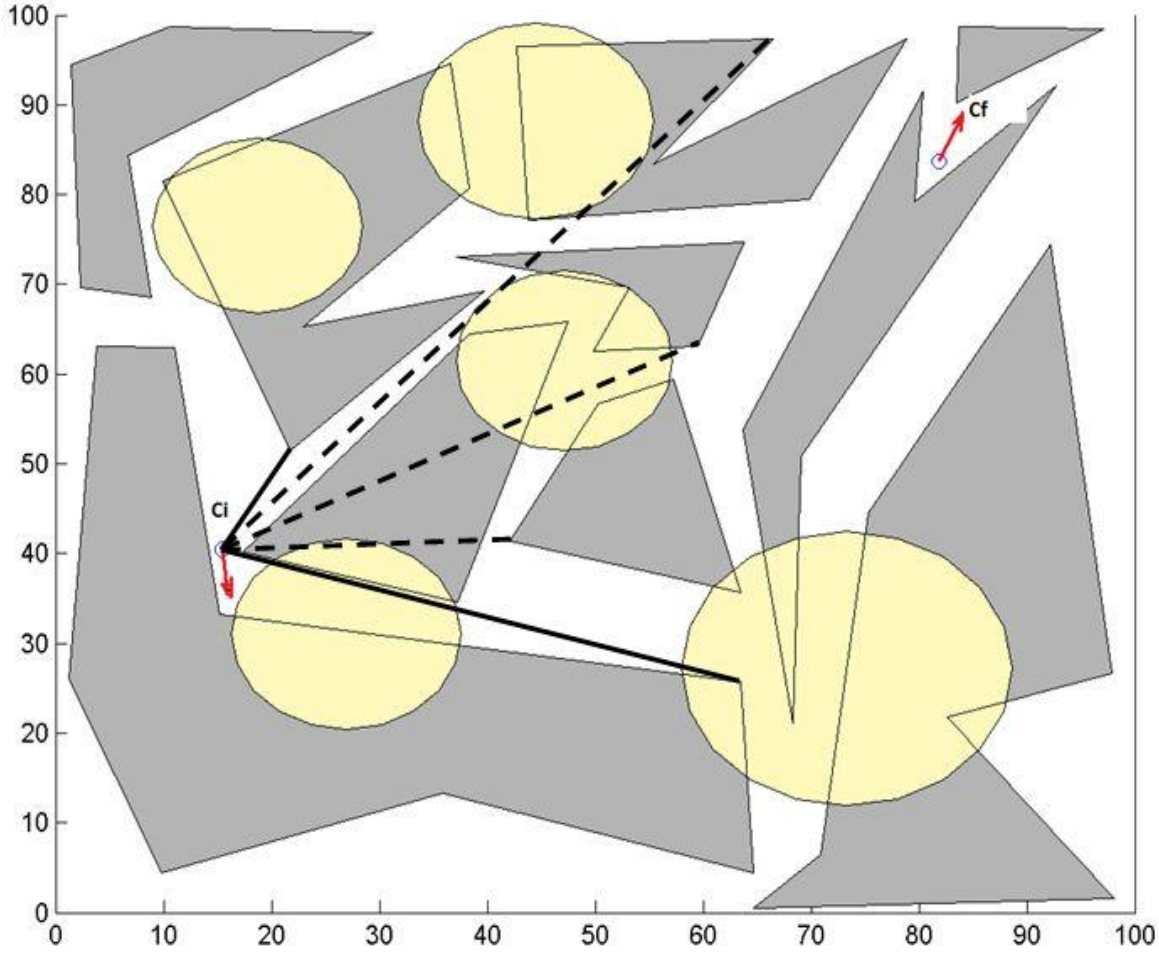


Figure 14 : Exemples de chemins réalisables (trait plein) qui ne traversent aucun obstacle et qui donnent donc naissance à un arc; exemples aussi de chemins non réalisables (en pointillé) qui traversent des obstacles et qui seront donc exclu du graphe.

L'ensemble de nœuds N est constitué par les configurations initiale et finale de l'avion et par les différentes configurations possibles associées aux sommets des obstacles. L'ensemble des arcs A est constitué par tous les arcs possibles, entre toutes les paires de nœuds de N qui ne passent pas par les obstacles. La Figure 14 illustre ces arcs. La définition de ces arcs, qui évitent les obstacles, est basée sur les travaux de Hormann et Agathos (2001) qui ont défini un algorithme à la base de la fonction 'inpolygon.m' dans

MATLAB. Cette fonction permet de détecter si un segment donné traverse un polygone quelconque.

Finalement, un nœud de G est un élément de l'espace configuration et un arc est tout arc allant de n'importe quel nœud pour joindre un autre sans traverser d'obstacles. Dans le graphe, un même point physique pourra correspondre à plusieurs nœuds (plusieurs configurations) n'ayant pas forcément les mêmes prédécesseurs ni les mêmes successeurs. La Figure 12 est un exemple de ce graphe. Par souci de simplification et de visibilité, tous les nœuds et les arcs ne sont pas représentés. Chaque point de la figure correspond à plusieurs nœuds – autant de nœuds que de directions possibles au niveau du point.

3.2.5.3 Élagage des nœuds que l'avion ne pourra pas visiter

Certains nœuds du graphe G, défini ci-dessus, ne sont pas utiles – le drone ne peut pas les visiter – et doivent être éliminés. Pour les sommets des obstacles toutes les configurations ne sont pas accessibles. La Figure 15 illustre des exemples de configuration que l'avion ne peut pas avoir lors de la visite du point en question. Lors de la visite du sommet a, dans cette figure, l'avion ne peut pas avoir une orientation dans l'intervalle $[\theta_1, \theta_2]$ (car cela veut dire qu'il aboutira dans l'obstacle, ce qui est impossible) ni dans l'intervalle $[\theta_1 + \pi, \theta_2 + \pi]$ (car il sort de l'obstacle). Donc tous les nœuds correspondants au sommet a avec ces directions interdites sont à éliminer du graphe G. Le sommet b, dans la même figure, ne peut pas être un point intermédiaire donc tous les nœuds lui correspondant dans le graphe G seront éliminés. En effet, quelle

que soit l'orientation de l'avion lors de l'approche de ce point, il ne pourra pas quitter la zone puisque l'angle est aigu. Tous les nœuds correspondant à des points dont l'amplitude du voisinage 'hors obstacle' est inférieure à 180° seront exclus du graphe. Car, compte tenu de la cinématique du drone, ces points sont des impasses.

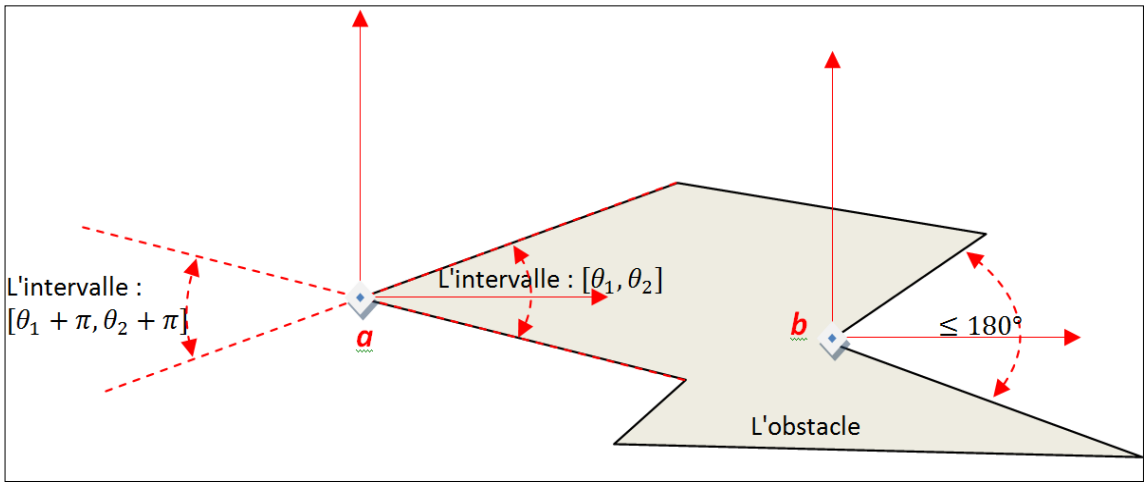


Figure 15 : Exemples de sommets d'obstacles. a est un sommet illustrant les directions que l'avion ne peut avoir à son voisinage. b est un sommet que l'avion ne peut visiter et qui est donc exclu du graphe.

3.2.5.4 L'intégration des zones de danger dans le graphe

Les zones de danger, comme dans la Figure 10, sont représentées par des cercles dont le rayon représente la portée du radar. Effectivement dans la réalité il existe des points où le risque d'interception par l'ennemi est nul, c'est ceux en dehors de la portée des radars. Notre graphe G, tel que présenté jusqu'à présent ne permet pas de tenir compte du contournement des zones de danger. Sur la Figure 16, nous avons un exemple de trajectoire avec un risque nul, sans passage par les zones de danger, dont notre graphe G ne tient pas compte (la courbe est dessinée sur la figure en tiret). En effet dans le cas de

cette figure, si nous ne considérons que les sommets des obstacles comme points intermédiaires nous ne pourrons jamais trouver une trajectoire de risque nul, car il y aura toujours une intersection avec les zones de danger, ni même plus petit qu'un certain seuil. Donc il est nécessaire d'introduire d'autres points intermédiaires dans G.

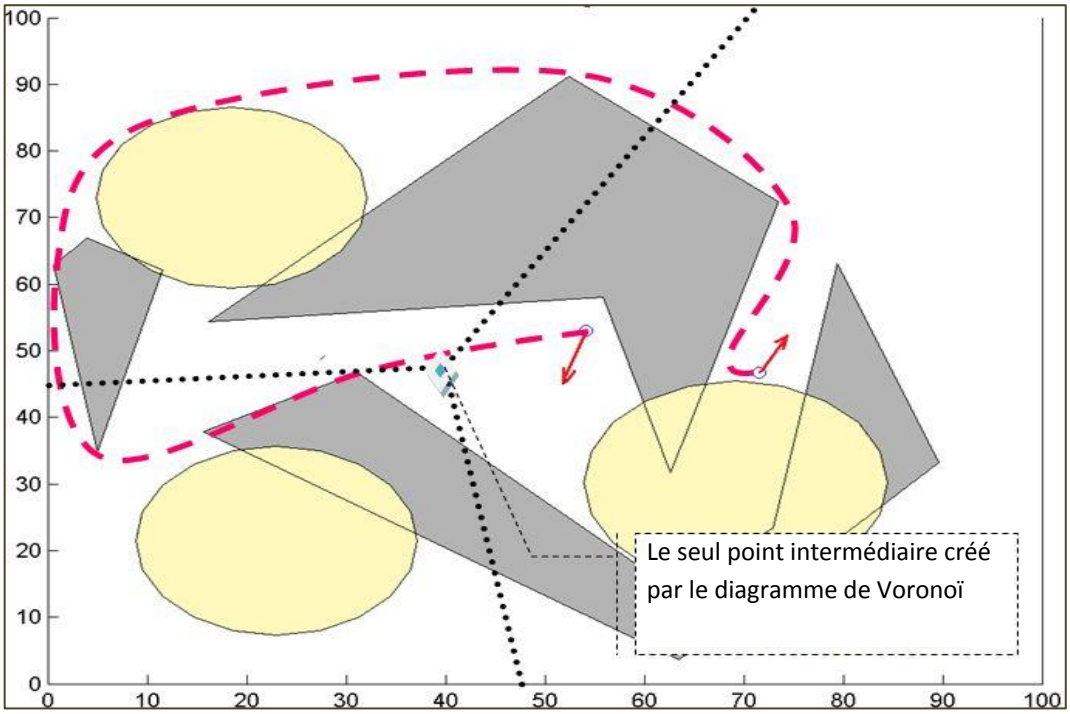


Figure 16 : Exemple de contournement des zones de danger sur un terrain où le diagramme de Voronoï est dessiné. Même le diagramme de Voronoï (traits en pointillé) ne permet pas le contournement des zones de danger.

Chandler (2001) utilise le diagramme de Voronoï pour le calcul de la trajectoire la moins risquée. Ce diagramme permet de trouver quelques points qui minimisent le risque d’exposition aux radars ennemis. Ceci est valide quand on ne connait pas la portée des radars car nous nous ne pouvons pas savoir comment les contourner. En outre, cette représentation ne permet pas assez de souplesse pour pouvoir nuancer plusieurs sensibilités au risque, voire interdire la traversée de la zone de danger. La Figure 16,

montre un diagramme de Voronoï. Il est représenté par les traits en pointillé (3 traits partant du centre de la figure) et un sommet, le point central de la figure. Ce diagramme ne permet pas de détecter le contournement (la courbe en tiret) des zones de danger.

Pour permettre ce contournement et la modélisation de plusieurs niveaux de risque, nous avons choisi de discrétiser la circonférence limite de la portée du radar pour pouvoir introduire des points intermédiaires permettant de contourner la zone de danger (voir Figure 17). Ceci constitue d'autres nœuds à intégrer dans le graphe G. Pour chaque configuration sur le contour de la zone de danger qui ne se trouve pas dans un obstacle nous ajoutons un nœud dans G. Ensuite nous créons des arcs, comme précédemment pour relier ces nœuds au reste du graphe.

Figure 17 : Illustration de l'intégration des zones de danger. Les points sur la circonférence représentent la discrétisation du contour de la zone de danger

3.2.5.5 Les contraintes cinématiques : les courbes de Dubins et le calcul des longueurs d'arcs

L'UAV est un robot non holonome. Sa principale contrainte cinématique est le fait qu'il ne peut changer de direction instantanément. Cette contrainte signifie qu'il existe un rayon de courbure minimum. Autrement dit, la variation de la direction θ de l'avion doit être confinée entre $-\phi_{max}$ et ϕ_{max}, l'angle de braquage maximum.

La résolution de cette problématique dans un terrain sans obstacle est faite par Dubins (1957) pour un véhicule qui ne peut pas faire marche arrière, ce qui est le cas de nos avions. Cette solution de Dubins, appelée 'Courbes de Dubins' est optimale pour un milieu sans zone de danger ni obstacles. D'autres travaux ont pu (Reeds et Shepp, 1990; Desaulniers, Soumis et Laurent, 1995) étendre cette solution à des véhicules capables de faire marche arrière mais ceci ne nous concerne pas.

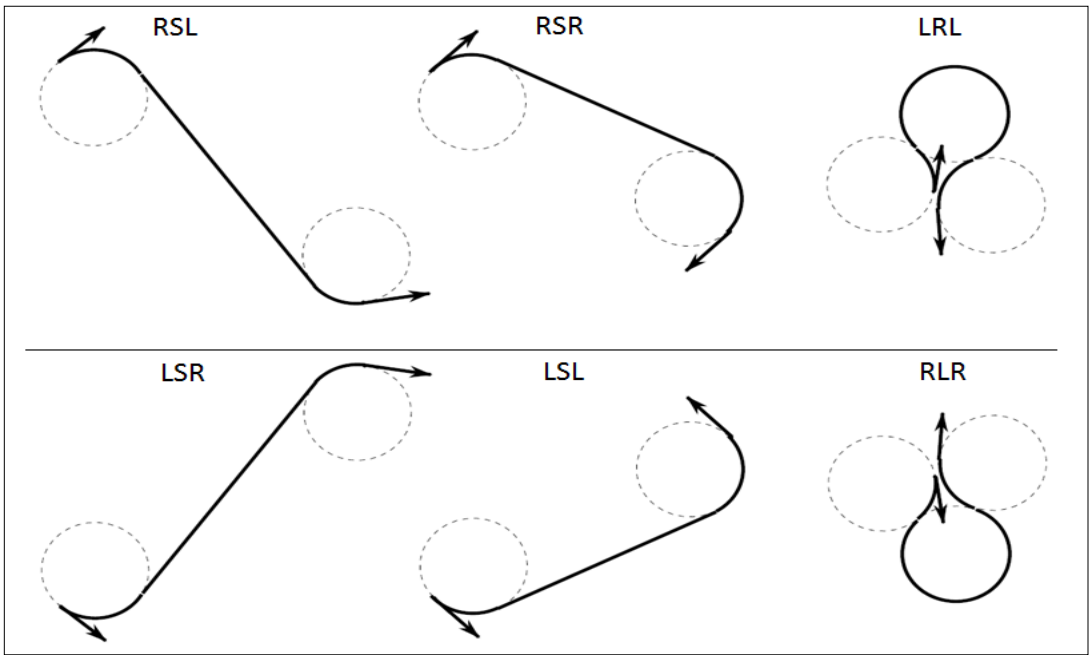


Figure 18 : Les six courbes de Dubins.

Si on considère un avion qui se déplace dans un espace de configuration donné avec un rayon minimum de courbure R, Dubins (1957) a établi que le plus court chemin non holonome entre deux configurations quelconques est toujours formé par la succession de 3 sous-chemins dans cet ordre :

- le 1er sous-chemin : un arc de cercle;
- le 2ème sous-chemin : un segment ou un arc;
- le 3ème sous-chemin : un arc de cercle.

Ce qui donne 6 possibilités de chemins à examiner entre deux configurations données. Considérons la notation suivante pour les sous-chemins :

- un segment est noté ‘S’;
- un arc de cercle de rayon R tournant vers la droite est noté ‘R’;
- un arc de cercle de rayon R tournant vers la gauche est noté ‘L’;

Les six possibilités définies par Dubins peuvent être représentées par l'un de ces six mots :

RSR – RSL – RLR – LSL – LSR – LRL

Le plus court chemin est l’un de ces six chemins (Figure 18). Dubins ne fournit pas la longueur de chacun des sous-chemins R, L, S. Ainsi pour trouver la plus courte trajectoire non holonome nous devons évaluer pour chacun des six chemins l'amplitude de chacun des sous-chemins le composant, en déduire la longueur totale de ce dernier, comparer les six longueurs et garder le plus court – voir exemple sur la Figure 19a.

Ce résultat est utilisé pour déterminer pour chaque arc du graphe G, sa première valence, la longueur du plus court chemin qui joint ses deux extrémités. Pour chaque arc de G, nous considérons les configurations correspondant aux nœuds origine et destination de l'arc. Nous calculons les six courbes de Dubins entre ces deux configurations et déterminons la plus courte. Nous associons ensuite la longueur de cette petite courbe à la longueur de l'arc. Pour ne pas refaire ces calculs, nous associons à chaque arc une matrice qui décrit cette plus courte courbe, C'est une matrice 2 par 3 qui décrit les 3 sous-chemins constituant cette courbe.

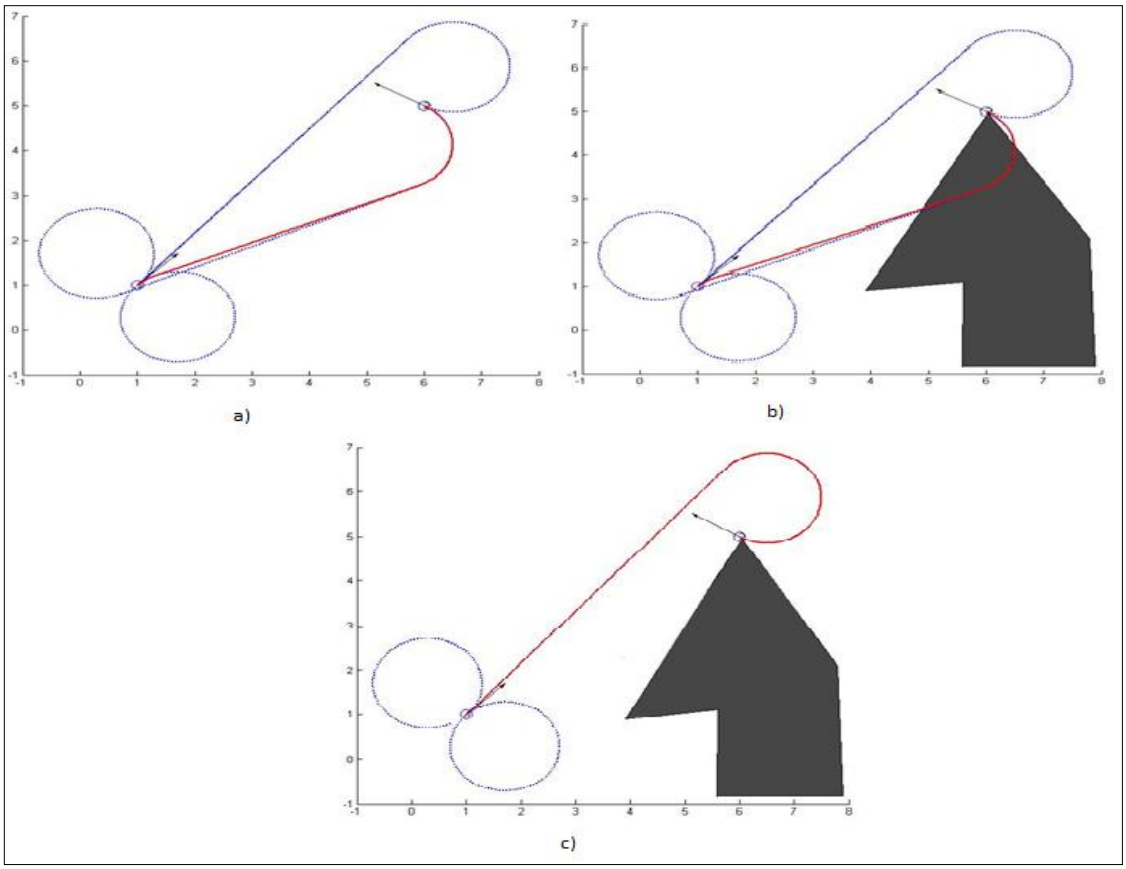


Figure 19 : Implémentation des courbes de Dubins. (a : sans obstacles; b : le résultat des courbes de Dubins propose une courbe qui traverse l'obstacle; c : notre implémentation évite l'obstacle).

3.2.5.6 Élagage des arcs non réalisables

Bien que le graphe de visibilité, introduit ci-dessus, permette de ne garder que les arcs réalisables c'est-à-dire ceux ne traversant pas les obstacles, la procédure de création des courbes de Dubins associées aux arcs peut induire des arcs non réalisables. Certaines de ces courbes, à cause des virages, touchent l'obstacle (Figure 19b). Le problème se pose au niveau des arcs dont l'un des nœuds correspond à un sommet d'un des obstacles (ce qui est la majorité des cas).

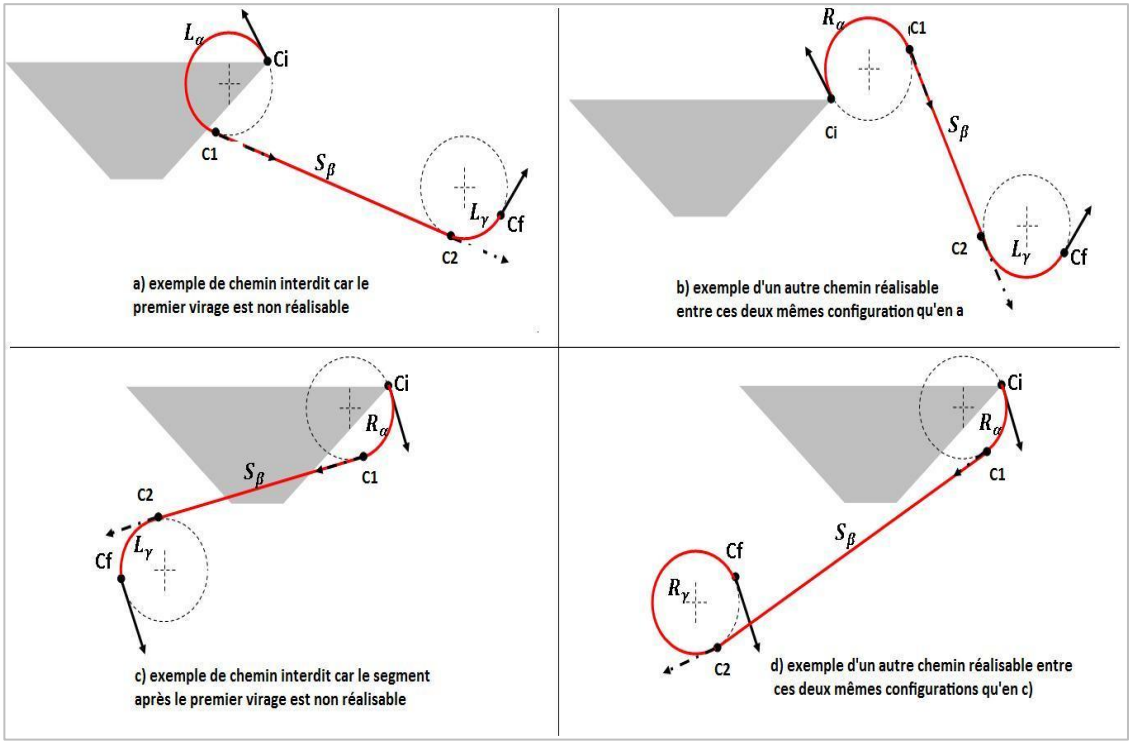


Figure 20 : Exemples de courbes de Dubins réalisables et non réalisables : a) la courbe LSL; b) la courbe RSL c)

On note qu'entre deux configurations données, les six courbes de Dubins ne sont pas toutes réalisables. La Figure 20 détaille l'analyse des certaines courbes (LSL, RSL, LSR, RSR) pour voir comment l'obstacle peut rendre certaines d'entre elles non réalisable. Il

existe un autre cas ou les courbes particulières (RLR, LRL) de Dubins ne sont pas réalisable c'est le cas où la distance entre les deux configurations en questions ne permet pas de faire trois arcs de cercle (la distance est trop grande). Cela nous a conduits à changer l'implémentation du calcul des courbes de Dubins. Au lieu de simplement évaluer les six courbes possibles et garder la plus courte, nous avons intégré une sous-routine qui permet pour chacune des sous courbes de vérifier si elle touche un obstacle ou non (Figure 20). Avec cette nouvelle implémentation des courbes de Dubins nous éliminons toutes les courbes non réalisables. Par suite le plus court chemin entre deux configurations, comme dans Figure 20c et Figure 20d, n'est pas forcément celui le plus court. Si les six courbes ne sont pas réalisables, à cause des obstacles, l'arc en entier est alors éliminé.

Cette nouvelle implémentation des courbes de Dubins est faite par l'exploitation de la fenêtre $[\theta_1, \theta_2]$, l'amplitude de la zone obstacle au voisinage du nœud en question pour détecter si l'avion ne rentre pas dans l'obstacle (Figure 15). Comme sur la Figure 20, entre les configurations Ci et Cf, on introduit deux autres configurations intermédiaires, $\mathrm{C1}$ et $\mathrm{C2}$. Puis entre chaque paire de ces quatre configurations on vérifie si le sous-chemin de Dubins passe par un obstacle. Pour les sous-chemins de type segment (Figure 20c) on utilise le même algorithme basé sur les travaux de Hormann et Agathos (2001). Pour les sous-chemins de type arc de cercle, Figure 20a, un calcul de changement de direction de l'avion permet de vérifier si cette dernière touche les fenêtres $[\theta_1, \theta_2]$ et $[\theta_1 + \pi, \theta_2 + \pi]$ et par suite si le sous-chemin passe ou non par l'obstacle.

3.2.5.7 Calcul du risque sur chaque arc

Maintenant que les nœuds et les arcs de notre graphe G sont définis et élagués pour ne garder que le nécessaire, il reste à finir le calcul des valences des arcs, le calcul du risque associé à chaque arc.

Pour le calcul du risque, on définit une fonction de densité du risque en chaque point du terrain, $r(x, y)$, qui est définie en fonction de la distance du point physique au radar. Alors pour chaque arc de notre graphe, le risque associé est l'intégrale de cette fonction densité sur l'arc. Donc le risque sera proportionnel, ou fonction, de la portion de la trajectoire qui se trouve dans une zone de danger (Figure 21).

La section qui suit 'Résolution et algorithme', présente le graphe obtenu par cette modélisation.

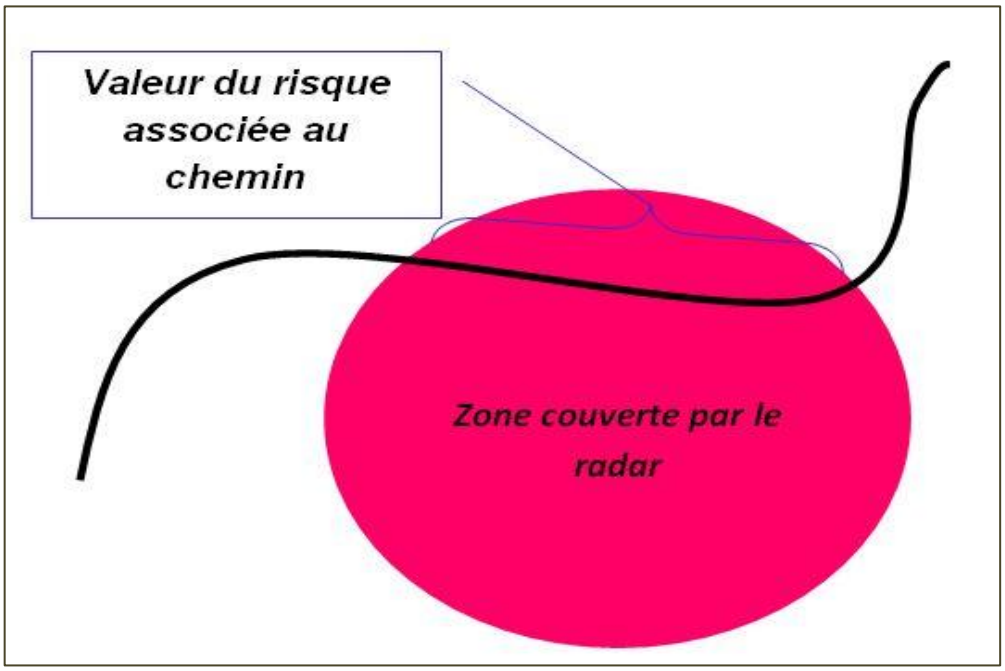


Figure 21 : Illustration du calcul du risque.

3.3 Résolution et Algorithme

Dans cette section nous présentons notre approche de résolution. Cette approche consiste en la modélisation du problème sous forme d'un graphe orienté sur lequel on résout un problème de plus court chemin classique avec l'intégration du calcul du risque dû au passage par des zones de danger. La présente section s'articule comme suit : la génération de graphe orienté est présentée en premier lieu avec une variante où la relaxation de la contrainte cinématique est introduite, suivie par les deux algorithmes de calcul de plus court chemin sur un graphe.

3.3.1 Génération du graphe orienté : graphe complet

Suite à la modélisation définie dans la section 3.2, on obtient un graphe orienté $G = (N, A)$. N, l'ensemble des nœuds est constitué de trois types de nœuds suivants :

- les configurations initiale et cible de l'avion;
- les configurations correspondant aux sommets des obstacles (à chaque sommet physique on associe plusieurs nœuds, autant que de directions que l'on peut définir au niveau de ce dernier, voir Figure 15);
- les configurations correspondant à la discrétisation des circonférences des zones de danger définies par les radars (à chaque point physique correspondent plusieurs nœuds, autant que la discrétisation définie sur la direction θ.

Chaque nœud de N est caractérisé par :

- (x, y) : le couple des coordonnées du point physique lui correspondant sur le terrain;
- θ : la direction, obtenue en discrétisant l'intervalle $[0, 2\pi]$;
- $[\theta_1, \theta_2]$: amplitude en radian désignant la zone obstacle adjacente à cette position, Figure 15 (ceci permet, lors de la discrétisation de la direction θ, de ne garder que les configurations réalisables, et lors du calcul d'un virage de déterminer si on touche ou non à l'obstacle, Figure 20).

A est l'ensemble des arcs. Entre chaque paire de nœuds de N, nous définissons deux arcs orientés, on évalue si chacun d'entre eux est réalisable ou non et si oui on l'ajoute à A. Les arcs sont à double valence, longueur et risque. Ils sont définis par les propriétés suivantes :

- la longueur : elle est évaluée par le calcul des courbes de Dubins entre ses deux extrémités;
- le risque : il est calculé en intégrant la densité du risque sur la courbe de Dubins associée;
- la courbe : la suite des sous-chemins de Dubins permettant d'aller de l'origine vers la fin de l'arc. C'est une matrice 2 par 3 qui décrit les 3 sous-chemins constituant cette courbe, la première ligne étant le type du sous-chemin (R, L ou S) et la deuxième ligne, la longueur du sous-chemin.

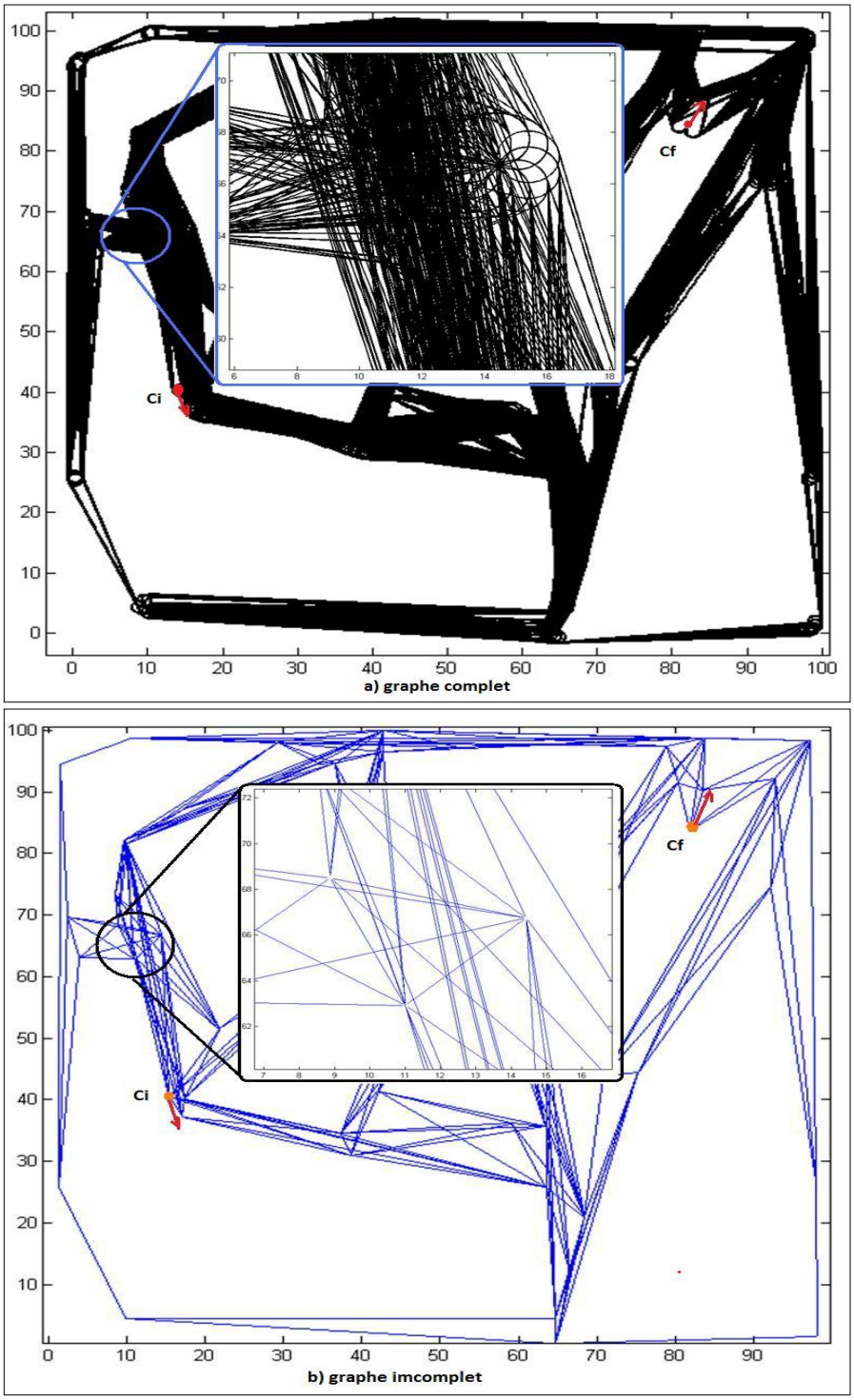


Figure 22 : Les graphes associés au terrain de la Figure 10 (a : le graphe complet, 771 nœuds et 3987 arcs; b : le graphe incomplet, 81 nœuds et 502 arcs)

L'algorithme 1 décrit la procédure de création de ce graphe qui contient plusieurs phases d'élagage pour rendre celui-ci le plus concis possible en ne gardant que les configurations nécessaires. La Figure 22a montre un exemple de graphe orienté généré par l'algorithme 1 et avec une subdivision en six de la direction θ . Il contient 771 nœuds et 3987 arcs. L'élagage a permis d'éliminer une centaine de nœuds et réduire le nombre d'arcs (dans notre cas le nombre d'arcs est divisé par sept).

L'algorithme 1 est une implémentation du graphe de visibilité et du calcul adapté des courbes de Dubins. Il s'articule comme suit : Dans un premier temps un calcul du graphe de visibilité (ses sommets sont ceux des obstacles et ses arcs sont toute arrête reliant les sommets des obstacles sans traverser ces derniers.) nous permet de créer juste les nœuds nécessaires qui potentiellement seront des points intermédiaires dans la trajectoire de l'avion (Figure 6). Ensuite, une implémentation du calcul des courbes de Dubins intégrée avec : 1) l'algorithme de Hormann et Agathos (2001) pour éviter les obstacles pour les lignes directes et 2) un calcul d'angle solide, de champs de vision pour éviter les obstacles lors de virage. Finalement une fonction de densité de risque pour déterminer le risque associé à chaque arc.

Le graphe orienté obtenu à l'issue de l'algorithme 1 est nommé graphe complet, car il tient compte de la non-holonomie en discrétisant la direction θ. Toutes les trajectoires calculées sur ce graphe correspondent à des trajectoires non holonomes, qui tiennent compte de la cinématique de l'avion, ce qui est une bonne qualité. Cependant la qualité des résultats des calculs sur ce graphe dépend grandement de la discrétisation de la direction θ, qui plus elle est fine, plus elle accroît la taille du graphe considéré. Pour

minimiser la taille du graphe, nous avons considéré la relaxation de la contrainte cinématique ce qui donnera lieu à un nouveau graphe orienté de plus petite taille car la direction θ ne sera pas discrétisée. Nous nommerons ce graphe, graphe incomplet.

Algorithme 1 : Génération du graphe complet orienté.

<u>Génération du graphe de visibilité (modélisation des obstacles)</u>

1- Création du nœud correspondant à la configuration initiale.

2- Création du nœud correspondant à la configuration finale.

3- Création des nœuds correspondant aux sommets des obstacles (modulo la discrétisation de la direction, chaque sommet d'obstacle génère plusieurs nœuds).

4- Élagage des nœuds dont la visite est impossible (Figure 15).

5- Création de deux arcs orientés entre chaque paire de nœuds créés jusqu'à présent.

6- Élagage des arcs non réalisables (touchent aux obstacles).

<u>Intégration des zones de danger</u>

7- Création des nœuds correspondants aux zones de danger modulo les différentes directions possibles (discrétisation de la circonférence de la zone et de la direction).

8- Élagage des nœuds dont la visite est impossible.

9- Connexion des derniers nœuds entre eux et aux autres nœuds via de nouveaux arcs.

10- Élagage des arcs non réalisables.

<u>Implémentation des courbes de Dubins et calcul de la longueur des arcs</u>

11- Pour chacun des arcs retenus.

12- Calculer les six courbes de Dubins entre les deux configurations correspondant à ses extrémités.

13- Élagage des courbes non réalisables.

14- Si au moins une courbe est réalisable.

15- Déterminer la plus courte et la longueur de l'arc est celle de cette dernière

16- Sinon

17- Éliminer l'arc, il est non réalisable

18- Calcul du risque associé aux arcs retenus

3.3.2 Relaxation de la contrainte de non-holonomie : génération du Graphe incomplet

L'idée de cette relaxation vient du fait que le graphe complet est énorme et la qualité du résultat dépend de la discrétisation de la direction θ. Dans notre situation, celle d'une mission de surveillance par drone, la distance entre obstacles est en général assez grande et permet donc la correction d'une trajectoire holonome en une trajectoire non holonome. La Figure 22 présente les deux graphes, complet et incomplet, pour un même terrain. Dans le graphe complet la direction est subdivisée en dix. Nous constatons bien, dans cet exemple, que la grande différence entre les tailles des deux graphes : le graphe complet a 771 nœuds et 3987 arcs alors que le graphe incomplet, 81 nœuds et 502 arcs. Cependant le graphe incomplet ne permet que le calcul de trajectoires holonomes qu'il faudra corriger. Ci-après, nous présentons la création de ce graphe et la procédure de correction.

3.3.2.1 Génération du Graphe incomplet

La procédure de création de ce graphe est la même que celle du graphe complet à l'exception que la direction n'est pas discrétisée. On utilise les mêmes ingrédients, graphe de visibilité, élagage des nœuds et des arcs non réalisables. Cependant, le calcul du risque et de la longueur sont approximés sur le segment rectilinéaire qui joint les extrémités de l'arc. L'algorithme 2 présente cette procédure.

3.3.2.2 Correction des trajectoires holonomes

Les figures, Figure 23 et Figure 24, illustrent cette correction. La trajectoire de la Figure 23 est une courbe holonome qui ne tient pas compte des contraintes cinématiques de

l'avion. Nous y remarquons des points anguleux A, B et C. Celle de la Figure 24 est cette même courbe après correction.

Algorithme 2 : Génération du graphe incomplet orienté.

<u>Génération du graphe de visibilité (modélisation des obstacles)</u>

1- Création du nœud correspondant à la configuration initiale.
2- Création du nœud correspondant à la configuration finale.
3- Création des nœuds correspondant aux sommets des obstacles (sans tenir compte de la direction).
4- Élagage des nœuds dont la visite est impossible.
5- Création de deux arcs orientés entre chaque paire de nœuds créés jusqu'à présent.
6- Élagage des arcs non réalisables.

<u>Intégration des zones de danger</u>

7- Création des nœuds correspondants aux zones de danger (discrétisation de la circonférence de la zone seulement).
8- Élagage des nœuds dont la visite est impossible.
9- Connexion de ces derniers nœuds entre eux et aux autres nœuds via de nouveaux arcs.
10- Élagage des arcs non réalisables.
11- Calcul de la longueur des arcs retenus comme étant la distance rectilinéaire séparant ses extrémités.
12- Calcul du risque associé aux arcs retenus.

La correction décrite dans l'algorithme 3 utilise deux éléments : 1) Pour chaque point intermédiaire de la courbe (les points A, B et C) on associe la meilleure direction possible pour créer une configuration (point physique avec orientation). Ici, nous avons privilégié la partie aval de la trajectoire : l'orientation d'un point intermédiaire de la courbe est celle lui permettant d'atteindre facilement le point suivant. Dans le cas de la Figure 24, la meilleure orientation au niveau du point A est celle donnée par le vecteur

$\overrightarrow{AB}$. 2) Pour la suite on utilise l'adaptation des courbes de Dubins, déjà expliquée avant, pour avoir le meilleur chemin réalisable en chaque deux points intermédiaires, et on calcule la longueur de la nouvelle trajectoire.

Cette correction ne fonctionne pas toujours. Par exemple, si une partie anguleuse de la trajectoire holonome (voir point A, Figure 23) se situe dans un couloir très serré avec des obstacles proches du point A de tous les côtés, la correction pourrait être impossible car l'espace serait insuffisant pour faire des virages. Dans notre cas, ce scénario est rare, voire inexistant, car la configuration des terrains réels de mission fait que les obstacles sont assez éloignés.

Algorithme 3 : Correction des trajectoires holonomes.

1- Pour chaque point intermédiaire de la trajectoire holonome
2- Calculer la meilleure direction à associer au point suivant
(ceci utilise le deuxième point suivant)
3- Calculer les six courbes de Dubins entre les deux configurations
(celle du point actuel et celle du suivant)
4- Élaguer les courbes non réalisables (qui touchent aux obstacles)
5- Si au moins une courbe est réalisable
6- Déterminer la plus courte associée à cet arc
7- La longueur de l'arc est celle de cette dernière
8- Sinon
9- La courbe n'est pas corrigeable.

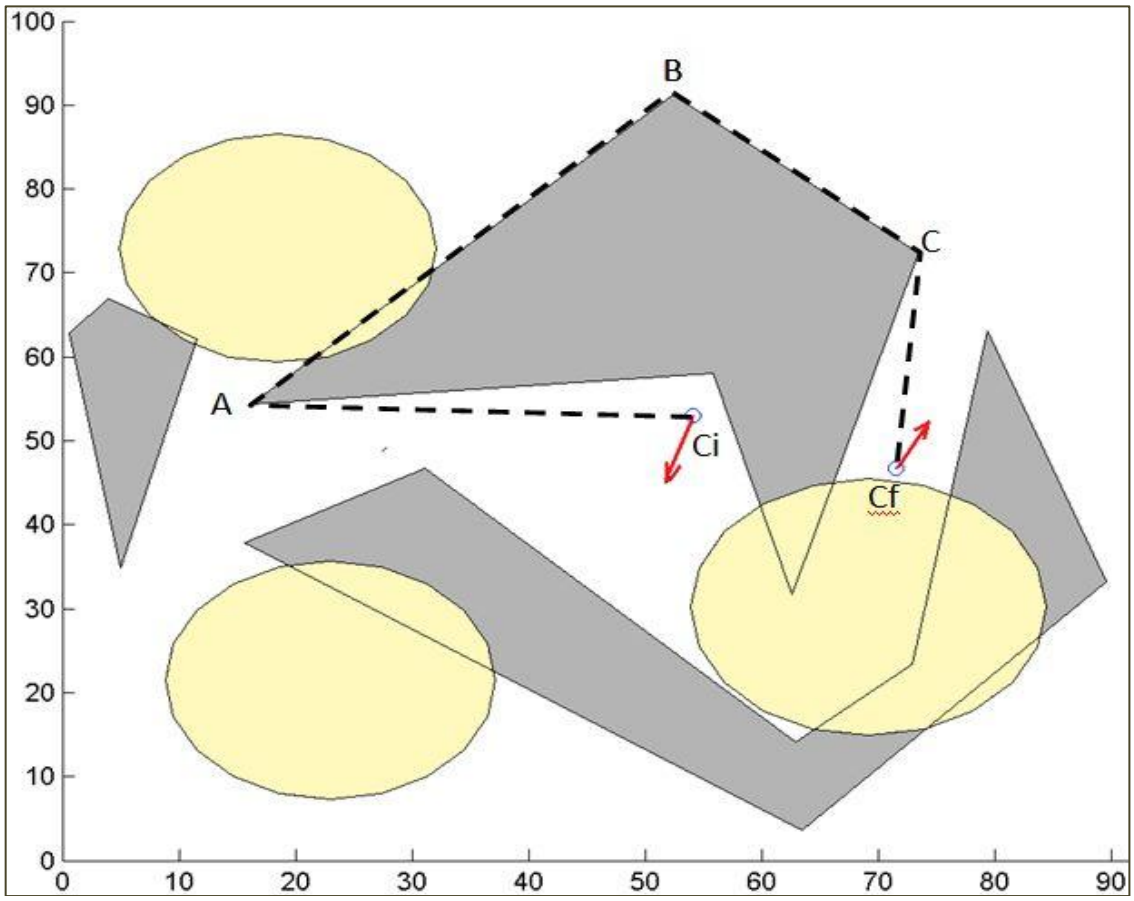


Figure 23 : Exemple de trajectoire holonome (issue du calcul sur le graphe incomplet, sans contrainte cinématique)

3.3.3 Calcul du plus court chemin : l'algorithme de Dijkstra

Cet algorithme permet de chercher le plus court chemin sur un graphe où la valence des arcs est positive. Dans notre cas une transformation du graphe était nécessaire, pour passer d'arcs à double valence vers des arcs à simple valence et pour inclure le risque dans l'objectif. Nous avons défini une autre longueur des arcs ($\alpha \in [0, +\infty]$) :

$$La\ longueur\ d'un\ arc = sa\ longueur + \alpha \times son\ risque$$

Après cette transformation l'implémentation est une implémentation classique de l’algorithme de Dijkstra.

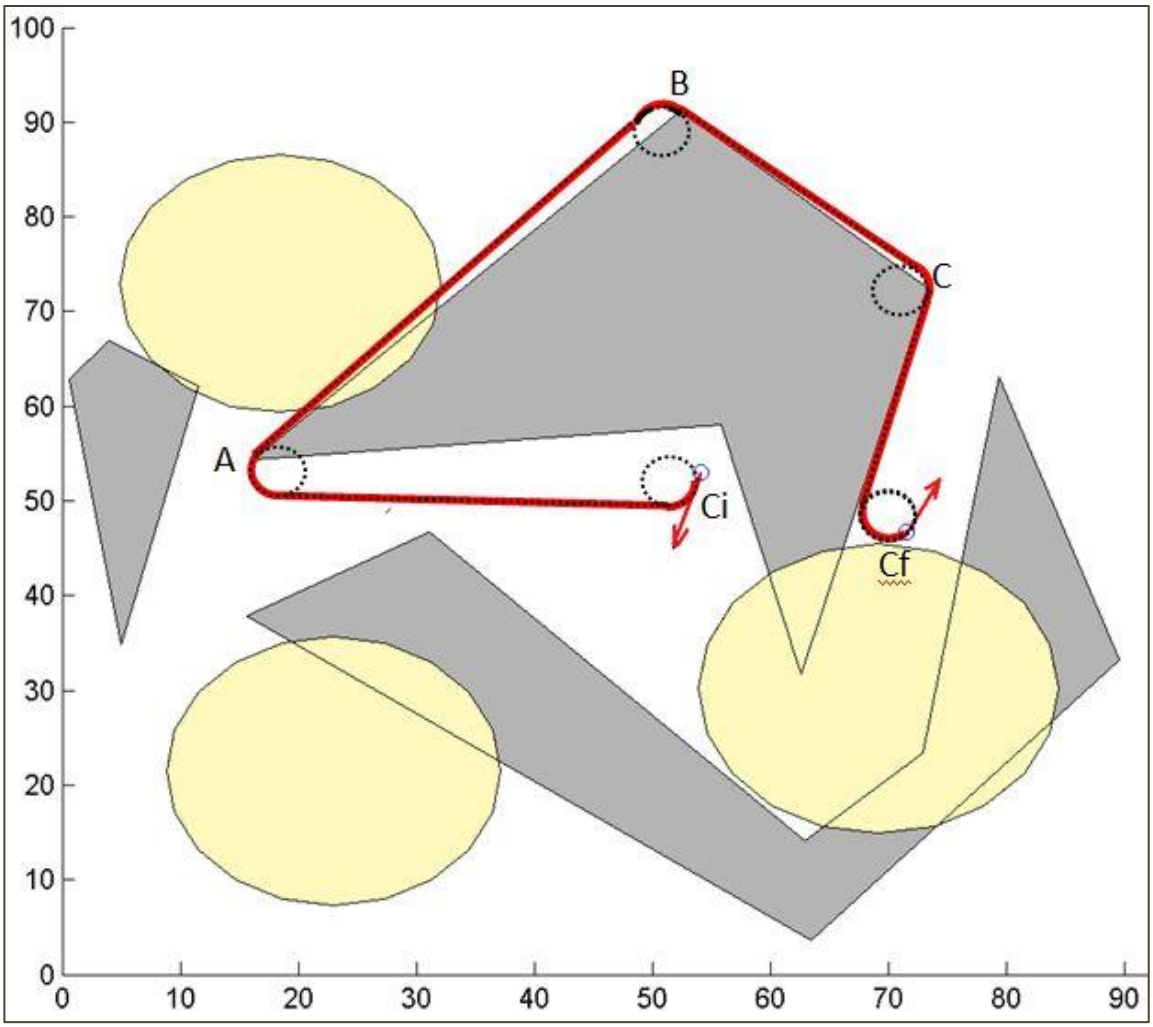


Figure 24 : La trajectoire corrigée, non holonome, correspondant à la correction de celle de la Figure 23.

3.3.4 Calcul du plus court chemin avec risque minimum : l'algorithme de Dijkstra modifié

Il s'agit du calcul d'un plus court chemin qui n'excède pas un certain risque limite. Ceci peut être modélisé comme un Plus Court Chemin avec Contraintes sur les Ressources

(PCC_CR). Dans notre cas la ressource est le risque, la quantité de ressource disponible est le risque limite et la demande se fait au niveau des arcs. Chaque arc a sa demande de risque égale au risque associé à ce dernier lors de la génération du graphe.

Algorithme 4 : Dijkstra avec risque limite (calcul du plus court chemin entre Ci et Cf).

Initialisation :

1- Initialiser la longueur du parcours et le risque à ∞ au niveau de tous les nœuds
2- Déclarer tous les nœuds comme non visités
3- Pour le nœud Ci,
4- la longueur du parcours = 0
5- le risque = 0

La boucle :

6- Tant qu'il reste des nœuds non visités,
7- Choisir parmi eux le nœud dont le parcours est le plus court (et le moins risqué si plusieurs candidats)
8- Déclarer ce nœud comme visité
9- Pour chacun des successeurs de ce nœud
10- Si le parcours traversant ce nœud est plus court et son risque est inférieur à la limite
11- Mettre à jour le parcours et le risque de ce successeur.

Dans la littérature, il existe plusieurs travaux sur le PCC_CR (Naghi et Soumis, 2000; Beasley et Christofides, 1989; Mehlhorn, 2000). Nous nous sommes inspirés de ces travaux pour le développement de notre algorithme 4.

3.4 Résultats et analyse

Cette section contient trois ensembles de tests. 1) Le premier concerne la discrétisation du terrain versus le graphe de visibilité. 2) Le deuxième concerne les résultats de notre approche avec la comparaison des deux formulations, le risque dans l'objectif ou dans la contrainte. 3) Et finalement la comparaison de la modélisation complète du terrain avec les contraintes cinématiques versus la relaxation de cette contrainte suivie d'une correction de la trajectoire.

La machine utilisée pour les tests est un ordinateur de type PC dont les caractéristiques sont :

- système d'exploitation : Windows XP;
- processeur : Intel double Core 2 GHz;
- mémoire vive : 4 Go;
- le logiciel : Matlab 2008.a.

Les tests utilisent des terrains générés sous Matlab. Nous avons développé sous Matlab une interface graphique qui permet de dessiner différents types de terrains avec des critères de nombre d'obstacles et de zones de danger. Pour le premier ensemble de tests, nous avons utilisé trois terrains au hasard car la configuration et le type du terrain ne sont pas importants. En effet, le but de cette partie est de voir l’impact de la taille du graphe généré via un maillage ou via graphe de visibilité sur la charge de calcul. Pour les deux autres ensembles de tests, nous avons généré sept terrains : Map01… Map07. Lors

de la création de ces terrains, nous avons couvert un grand nombre de types de terrain en termes de nombre d'obstacles, de zones de danger et de leur configuration (Tableau 1).

Tableau 1 : Descriptions des terrains

Terrain	Nb Obstacles	Nb Dangers	Temps pour la génération du Graphe (en secondes)	Nb Nœuds	Nb Arcs
Map01	1	1	9.036	127	417
Map02	1	1	10.206	121	622
Map03	4	3	78.095	345	2274
Map04	4	3	89.189	381	2344
Map05	5	4	131.646	512	2576
Map06	10	5	262.914	771	3987
Map07	10	5	513.757	761	9818

3.4.1 Discrétisation du terrain versus graphe de visibilité

Le but de ces tests est de prouver que le maillage du terrain n'est pas nécessaire et qu'il demande plus de charges de calcul. Tous les temps indiqués dans les tableaux qui suivent sont en secondes. Ici nous avons utilisé trois terrains différents – numérotés 1, 2 et 3 – pour lesquels on a modélisé le terrain en discrétisant comme suggéré dans (De Mot et al., 2002; Kim et Hespanha, 2003) – la méthode 1 – et puis en utilisant notre approche basée sur le graphe de visibilité – la méthode 2.

D'après le Tableau 2, on voit bien que la méthode 2, notre méthode basée sur le graphe de visibilité, est plus rapide pour la génération du graphe. Le graphe obtenu par le graphe de visibilité est de plus petite taille ce qui explique que le calcul de la trajectoire sur ce dernier est plus rapide, comme indiqué sur le Tableau 4. Ce dernier tableau montre aussi qu'on obtient les mêmes résultats, mêmes trajectoires, par les deux méthodes. La conclusion est que nous n'avons pas besoin des autres points du plan

(points encerclés sur la Figure 13) pour calculer la trajectoire optimale. La trajectoire optimale est soit une courbe, de Dubins, joignant directement le départ à la cible, soit une courbe qui est tangente aux obstacles. Le maillage du terrain n'est pas nécessaire et n'apporte rien de plus à la qualité de la solution.

Tableau 2 : Comparaison des temps de génartion des graphes par discrétisation versus graphe de visibilité.

Terrain	Méthode	Temps pour la génération du graphe (en secondes)	Nb Nœuds	Nb Arcs
1	2	18.750	48	640
1	1	137.000	74	1876
2	2	15.625	45	558
2	1	172.188	76	2050
3	2	17.391	45	616
3	1	145.176	69	1900

Notre méthode est plus rapide mais que le temps de génération du graphe est toujours très long. Regardons plus en détail le temps de génération du graphe par notre méthode (la méthode 2) pour le terrain 3. Dans le Tableau 3, nous constatons que c’est l’évitement des obstacles qui prend le plus de temps 83 % du temps total de la génération du graphe. Par contre si nous considérons que notre terrain ne contient que des obstacles convexes, nous remarquons dans le Tableau 3 que le temps chute beaucoup (80 %), il passe de 145 ses à 28 sec. Et ce n’est plus l’évitement des obstacles qui coûte le plus cher mais c’est les courbes de Dubins (82 % du temps total).

Tableau 3 : Analyse détaillée du temps de génération du graphe pour le terrain 3

avec des obstacles non convexes et puis avec obstacles convexes

		Génération des nœuds		Génération des arcs			Total
		noeuds obstacles	noeuds dangers	Évitement des obstacles	courbes de Dubins	Calcul du risque	
Obstacles non convexes	en sec	0.184	0.169	**120.377**	23.255	1.192	145.176
	en %	0.13 %	0.12 %	**82.92 %**	16.02 %	0.82 %	100 %
Obstacles convexes	en sec	0.171	0.188	3.255	**22.917**	1.461	27.992
	en %	0.61 %	0.67 %	11.63 %	**81.87 %**	5.22 %	100 %

Tableau 4 : Comparaison des résultats obtenus sur les différents graphes des deux méthodes.

Nom Graphe	Poids du Risque	PCC	Valeur du PCC	Risque Total	Temps pour le Calcul (en sec)
Terrain3_Methode1	0,1	[1 11 2]	17,102	5,4406	249,33
Terrain3_Methode1	1	[1 26 45 2]	17,487	0,0845	212,59
Terrain3_Methode1	10	[1 26 25 2]	17,647	0	202,39
Terrain3_Methode2	0,1	[1 11 2]	17,102	5,4406	16,59
Terrain3_Methode2	1	[1 26 45 2]	17,487	0,0845	17,66
Terrain3_Methode2	10	[1 26 25 2]	17,647	0	16,86

3.4.2 La comparaison des deux formulations

Pour cette partie nous avons utilisé sept différents terrains générés sous Matlab. Ces terrains représentent des situations plus au moins compliquées, avec plus ou moins d'obstacles, obstacles proches ou éloignés. Ces terrains sont joints dans l'annexe 2. Le Tableau 1 donne la description de ces terrains en termes de nombre de zones de danger et de nombre d'obstacles. Les graphes sont générés par notre méthode, basée sur le graphe de visibilité, en tenant compte de toutes les contraintes et en générant un graphe

complet. Le Tableau 1 indique pour chaque terrain la taille du graphe complet correspondant (nombre de nœuds et nombre d'arcs).

Tableau 5 : Résultat par les deux formulations pour le terrain Map06. On remarque que les deux formulations sont équivalentes.

	Poids du Risque	PCC	Valeur du PCC	Risque du PCC	Temps pour le Calcul (en sec)
Risque dans l'objectif	0,01	1 86 16 10 191 2	106.5748	30.627	0.21677
	0,1	1 86 16 10 191 2	106.5748	30.627	0.21147
	1	1 103 82 50 10 191 2	**114.6041**	**22.526**	0.21349
	10	1 100 588 584 573 640 186 187 2	141.0835	0	0.14674
	100	1 100 588 584 573 640 186 187 2	141.0835	0	0.13
Risque dans la contrainte	100	1 86 16 10 191 2	106.5748	30.627	0.22052
	50	1 86 16 10 191 2	106.5748	30.627	0.20958
	5	1 599 136 577 573 640 186 187 2	**140.9467**	**3.757**	0.14427
	1	1 100 588 584 573 640 186 187 2	141.0835	0	0.13032
	0,5	1 100 588 584 573 640 186 187 2	141.0835	0	0.13285

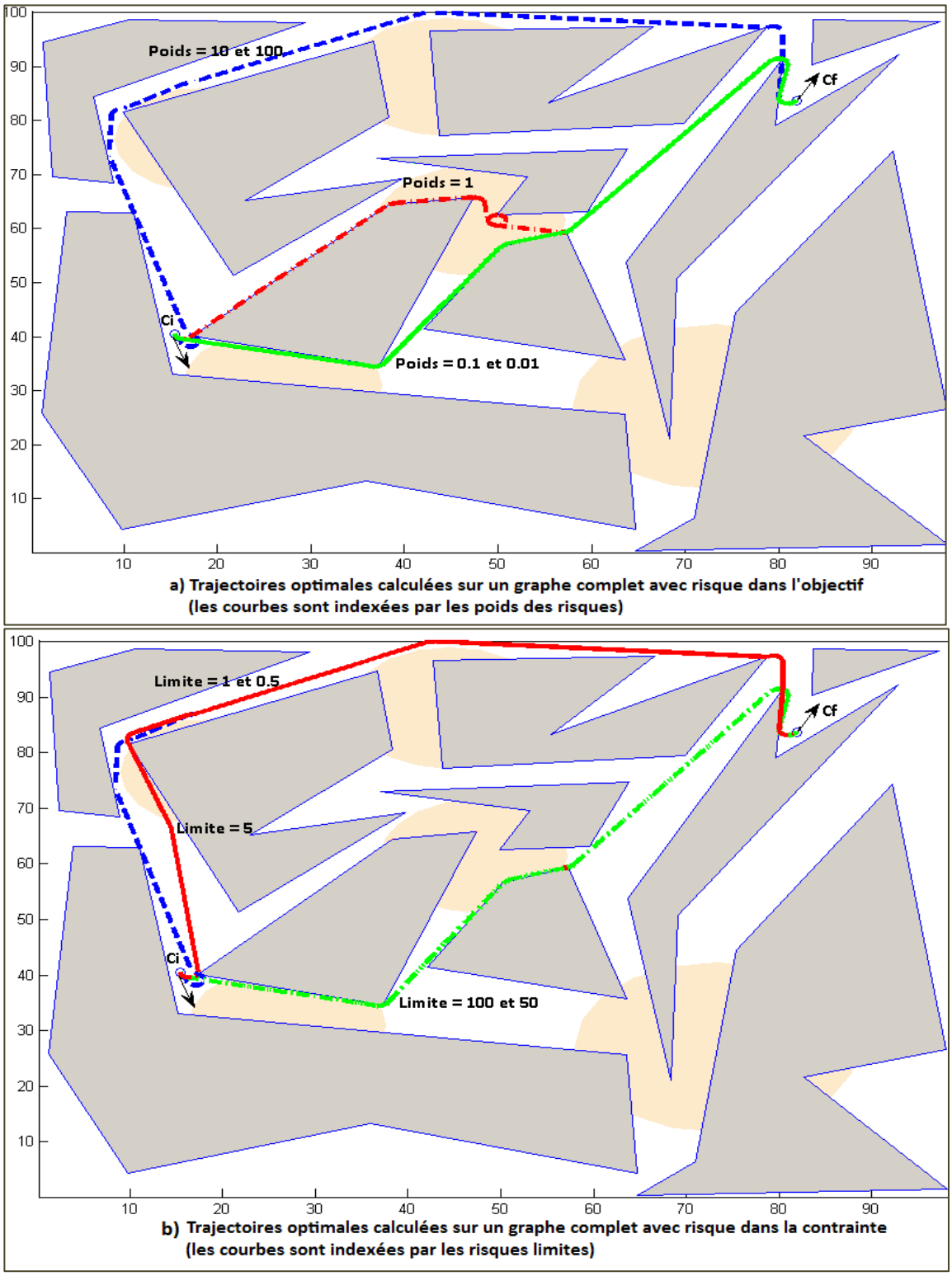


Figure 25 : Trajectoire optimale obtenue sur le terrain Map06 selon les deux formulations.

Le Tableau 5 présente pour l'exemple du terrain 'Map06' les résultats selon chacune des deux formulations. Ce tableau présente la trajectoire obtenue, sa longueur, la valeur du risque sur cette trajectoire et le temps de calcul en seconde pour sa détermination. La Figure 25 illustre ces trajectoires sur le terrain.

Une première lecture du Tableau 5 montre que ces deux formulations sont identiques dans leur logique de fonctionnement : un poids du risque élevé donne le même type de résultat qu'un risque limite très bas. Cependant, sur ce même tableau, pour un poids de 1 et un risque limite de 5 nous n'avons pas exactement la même valeur du chemin et du risque, contrairement aux autres valeurs. Ceci est dû au fait que la valeur 5 n'est pas la limite équivalente au poids 1. "Déterminer quelle limite de risque donne exactement le même objectif qu'un poids donné ou inversement?" est une question intéressante mais nous ne la traiterons pas ici car notre objectif est simplement de montrer que les deux formulations se comportent de la même façon.

En conclusion, de ces résultats, on constate que les deux formulations donnent les mêmes résultats du moment que le seuil du risque limite est équivalent au poids utilisé dans la première formulation. Les trajectoires sont identiques excepté quelques cas. Le temps de calcul des deux implémentations est quasiment la même excepté un léger et petit avantage pour la formulation 1 avec le risque dans l’objectif.

Sur la Figure 25a, la courbe associée au 'poids = 1' présente une anomalie, la boucle au milieu. Ceci est dû à la limite sur notre discrétisation de la direction θ. Une discrétisation

plus fine – qui demandera plus de temps de calcul – éliminera cette erreur. Ceci sera discuté dans la sous-section suivante.

3.4.3 La relaxation de la contrainte cinématique

Avec les mêmes terrains définis dans la section ci-dessus, nous avons généré les deux types de graphes complet et incomplet décrits dans la section 3.3. Le graphe complet est celui obtenu en intégrant toutes les contraintes, le graphe incomplet est celui obtenu en relaxant la contrainte cinématique. Lorsqu'on néglige cette contrainte, la discrétisation de la direction θ n'est plus nécessaire. Ceci réduit grandement la taille du graphe, le temps de génération de ce dernier et le temps de calcul de la trajectoire qui par suite est holonome. Ce graphe incomplet nécessite un posttraitement de la trajectoire, une correction de cette dernière pour la rendre non holonome (voir sous-section 'Correction des trajectoires holonomes'). Toutes les trajectoires holonomes ne peuvent cependant pas être corrigées! Dans notre cas celui des avions, ceci ne pose pas de problème car la distance entre les obstacles est assez grande et permet toujours une correction de la trajectoire. Dans d'autres contextes où l'espace entre les obstacles est très réduit cela pourrait être impossible.

Le Tableau 6 présente les résultats des deux modélisations. La Figure 22 illustre la différence entre les deux graphes pour le cas du terrain Map06. Le gain pour la génération du graphe, en termes de temps pour la modélisation du terrain, est énorme. Le rapport de réduction du temps de calcul entre la génération d'un graphe complet et celui

d'un graphe incomplet est supérieur à 10. Regardons maintenant l'impact sur le calcul de la trajectoire. Nous prendrons comme exemple celui du terrain Map06.

Tableau 6 : Comparaison de la génération d'un graphe complet (avec la contrainte de non-holonomie) versus un graphe incomplet (sans la contrainte de non-holonomie).

Terrain	**Graphe complet**			**Graphe incomplet**		
	Nb Nœuds	Nb Arcs	Temps pour la génération du graphe	Nb Nœuds	Nb Arcs	Temps pour la génération du graphe
Map01	127	417	9.036	15	54	0.90942
Map02	121	622	10.206	14	82	0.64772
Map03	345	2274	78.095	36	262	5.0275
Map04	381	2344	89.189	40	270	5.8922
Map05	512	2576	131.646	54	292	9.0658
Map06	771	3987	262.914	81	502	19.4589
Map07	761	9818	513.757	81	1072	49.8323

Tableau 7 : Comparaison des deux modélisations (des deux graphes) avec la formulation 1 (risque dans l'objectif).

	Poids du Risque	**PCC Val après (et avant) correction**	**PCC Risque**	**Temps pour le calcul du chemin**	**Temps pour la correction**	**Total**
Graphe incomplet	0,01	105,1188 (99,22)	21,9882	0,10521	0,27625	0,3815
	0,1	106,4828 (99,51)	0	0,025359	0,045138	0,0705
	1	106,4828 (99,51)	0	0,017966	0,032927	0,0509
	10	106,4828 (99,51)	0	0,013964	0,034538	0,0485
	100	106,4828 (99,51)	0	0,015284	0,033666	0,0490
Graphe complet	0,01	106,5748 (- -)	30,627	0,21677	0	0,2168
	0,1	106,5748 (- -)	30,627	0,21147	0	0,2115
	1	114,6041 (- -)	22,5263	0,21349	0	0,2135
	10	141,0835 (- -)	0	0,14674	0	0,1467
	100	141,0835 (- -)	0	0,13	0	0,1300

D'après les tableaux, Tableau 7 et Tableau 8, et les figures, Figure 25 et Figure 26, on constate que la relaxation accélère le calcul de la trajectoire aussi (voir colonne total); et ce pour les deux formulations. De plus les résultats sont de meilleure qualité que ceux obtenus avec le graphe complet. Ceci est dû à la limitation au niveau de la discrétisation de l'angle θ. Sur la Figure 26 (résultats avec un graphe incomplet) nous avons une meilleure trajectoire (celle en pointillé sur la Figure 26, elle est associé au risque limite 5 dans la Figure 26b et au poids 1 dans la Figure 26a) la plus courte et sans risque. Dans la modélisation avec graphe complet, Figure 25, nous ne trouvons pas cette solution, pour l'avoir il faudrait affiner la discrétisation de la direction θ. Dans notre exemple une discrétisation en 100 valeurs de la direction (ce qui multiplie la taille du problème de la Figure 25 par 10) permet d'avoir cette même solution.

Tableau 8 : Comparaison des deux modélisations (des deux graphes) avec la formulation 2 (risque dans la contrainte).

	Risque limite	PCC Val après (et avant) correction	PCC Risque	Temps pour le calcul du chemin	Temps pour la correction	Total
Graphe incomplet	100	105,1088 (99.22)	21,9882	0,018946	0,039043	0,0580
	50	105,1088 (99.22)	21,9882	0,018452	0,057801	0,0763
	5	106,4828 (99.51)	0	0,014079	0,032608	0,0467
	1	106,4828 (99.51)	0	0,014579	0,033729	0,0483
	0,5	106,4828 (99.51)	0	0,093619	0,27042	0,3640
Graphe complet	100	106,5748 (- -)	30,627	0,22052	0	0,2205
	50	106,5748 (- -)	30,627	0,20958	0	0,2096
	5	140,9467 (- -)	3,757	0,14427	0	0,1443
	1	141,0835 (- -)	0	0,13032	0	0,1303
	0,5	141,0835 (- -)	0	0,13285	0	0,1329

La relaxation de la contrainte de cinématique a beaucoup d'avantages : le graphe est plus petit, ce qui réduit le temps de calcul et on évite également le problème de la boucle, comme celle détectée dans la Figure 25a. La relaxation permet de bien trouver toutes les bonnes opportunités de réduction de la longueur de la trajectoire. Cependant si la distance entre les obstacles est très petite cette relaxation ne nous garantit pas une bonne solution et on risque même de générer des trajectoires qu'on ne peut pas corriger!

Par contre, la modélisation incluant cette contrainte, celle du graphe complet, a un défaut majeur : le temps de calcul est plus important, principalement le temps pour la génération du graphe. Néanmoins, avec une bonne finesse de la discrétisation de la direction θ on aura les meilleurs résultats si on est prêt à sacrifier beaucoup de temps dans la génération du graphe. Dans notre cas, celui de la planification d'une mission, la modélisation du terrain en graphe ne se fait qu'une fois lors de la préparation de la mission. Donc nous pouvons nous permettre une grande charge de calcul pour la préparation du terrain du moment qu'on dispose d'assez de mémoire pour stocker le graphe.

Aussi, et de façon globale, nous remarquons que plus on augmente notre tolérance au risque, plus la procédure nous propose des solutions qui traversent les zones de danger. Ceci est vrai quelque soit la formulation ou le type de graphe utilisé.

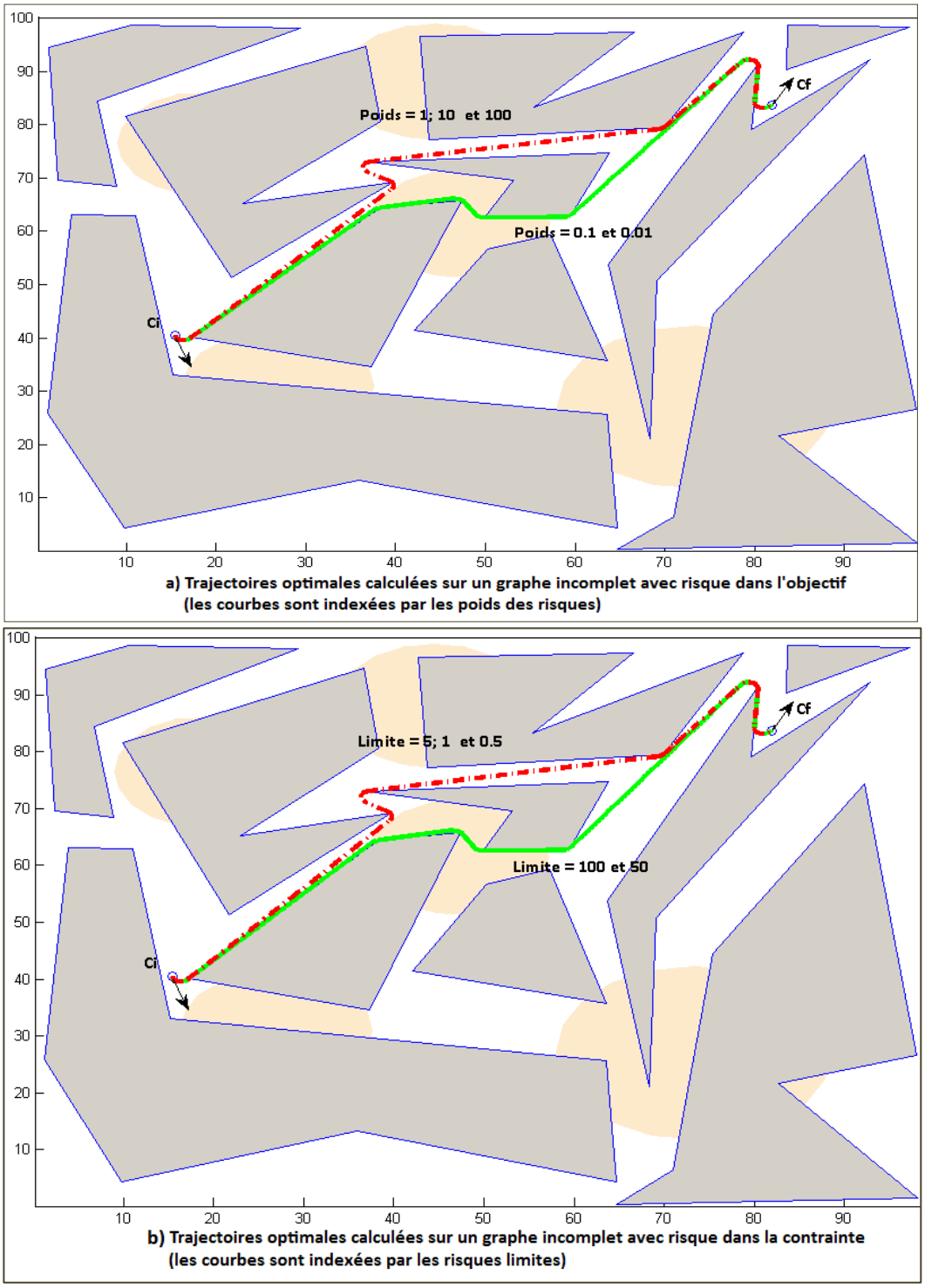


Figure 26 : Trajectoire optimale obtenue sur le terrain Map06 selon les deux formulations et en utilisant un graphe incomplet (c.-à-d. en relaxant la contrainte cinématique).

3.5 Conclusion

La conclusion de ce chapitre est que nous n'avons pas besoin des points du plan encerclés sur la Figure 13 pour calculer la trajectoire optimale. La trajectoire optimale est soit une courbe, de Dubins, joignant directement le départ à la cible, soit une courbe qui est tangente aux obstacles. Le maillage du terrain n'est pas nécessaire et n'apporte rien de plus à la qualité de la solution. Nous remarquons aussi que les obstacles non convexes augmentent le temps de génération du graphe de 80 %

De plus les résultats numériques indiquent que les deux formulations sont équivalentes du moment que le seuil du risque limite est équivalent au poids utilisé dans la première formulation. Les trajectoires sont identiques excepté quelques cas. La performance des deux implémentations est quasiment la même excepté un léger et petit avantage de la formulation 1 avec le risque dans la contrainte.

Un autre résultat important est que la modélisation avec le graphe incomplet permet de gagner énormément de temps de calcul sans perdre en qualité, on y gagne même dans certains cas. Cependant ce graphe nécessite une correction de la trajectoire qui n’est pas réalisable pour tous les types de problèmes.

CHAPITRE 4

PLANIFICATION DE MISSION D'UNE FLOTTE DE DRONES NON HOLONOMES DANS UN ENVIRONNEMENT À RISQUE ET EN PRÉSENCE D'OBSTACLES

Le but de ce chapitre est répondre à la question de la planification de mission de surveillance d'un terrain connu d'avance. Cette surveillance est faite par une flotte de drones dans un terrain plein d'obstacles, et on cherche à couvrir toutes les cibles en minimisant le coût et la durée de la mission tout en respectant l'autonomie des avions et sans dépasser un risque limite.

Nous formulons cette problématique comme un problème de tournées de véhicules avec capacités (CVRP) sur les arcs. Premièrement, le terrain est modélisé via un graphe orienté (voir chapitre 3) qui tient compte de toutes les contraintes. À partir de ce dernier, on génère un multi-graphe pour répondre aux contraintes de capacité sur les arcs (contraintes d'autonomie et du risque). La résolution se fait par une métaheuristique Tabou à double liste taboue sur ce multi-graphe.

Nous commençons par énoncer le problème, son objectif, ses contraintes, ainsi que les hypothèses de notre recherche. La formulation en CVRP est ensuite présentée. Finalement, les éléments du Tabou utilisé sont détaillés et on présente les résultats.

4.1 Le problème : la planification d'une mission de surveillance

Le chapitre 1 expose largement la problématique avec tous ses aspects. Il existe deux types de missions, soit de surveillance ou de reconnaissance. Elles sont fondamentalement différentes l'une de l'autre à plusieurs niveaux : la connaissance du terrain, la définition du but de la mission et des cibles, les exigences en termes de données à collecter et de trajectoire à faire, le budget.

Ici on s'intéresse à une mission de surveillance (voir chapitre 1). Nous connaissons les données du terrain, les cibles et les zones de danger (sous contrôle ennemi). Cette problématique est très vaste car elle peut avoir plusieurs objectifs et peut se décliner de différentes façons. Les différentes variantes de notre problématique sont dues au contrôle des UAVs, leur type, l'objectif à atteindre. Il est également important de savoir s'il s'agit d'une planification (généralement, un problème statique) ou d'une exécution (généralement, un problème dynamique) de mission. Une autre dimension à inclure est la dimension stochastique, toutes les données ne sont pas nécessairement connues d'une manière déterministe.

Au sein de toutes ces variantes, la problématique qui nous intéresse se situe comme suit : **on a un nombre fixe d'UAVs et on cherchera à planifier leur mission de reconnaissance d'un terrain, plein d'obstacles et de zones de danger, en vue de**

visiter toutes les cibles connues d'avance. L'objectif est de minimiser le coût et la durée de la mission tout en respectant l'autonomie des véhicules et en gardant le niveau de risque de la mission inférieur à un certain seuil.

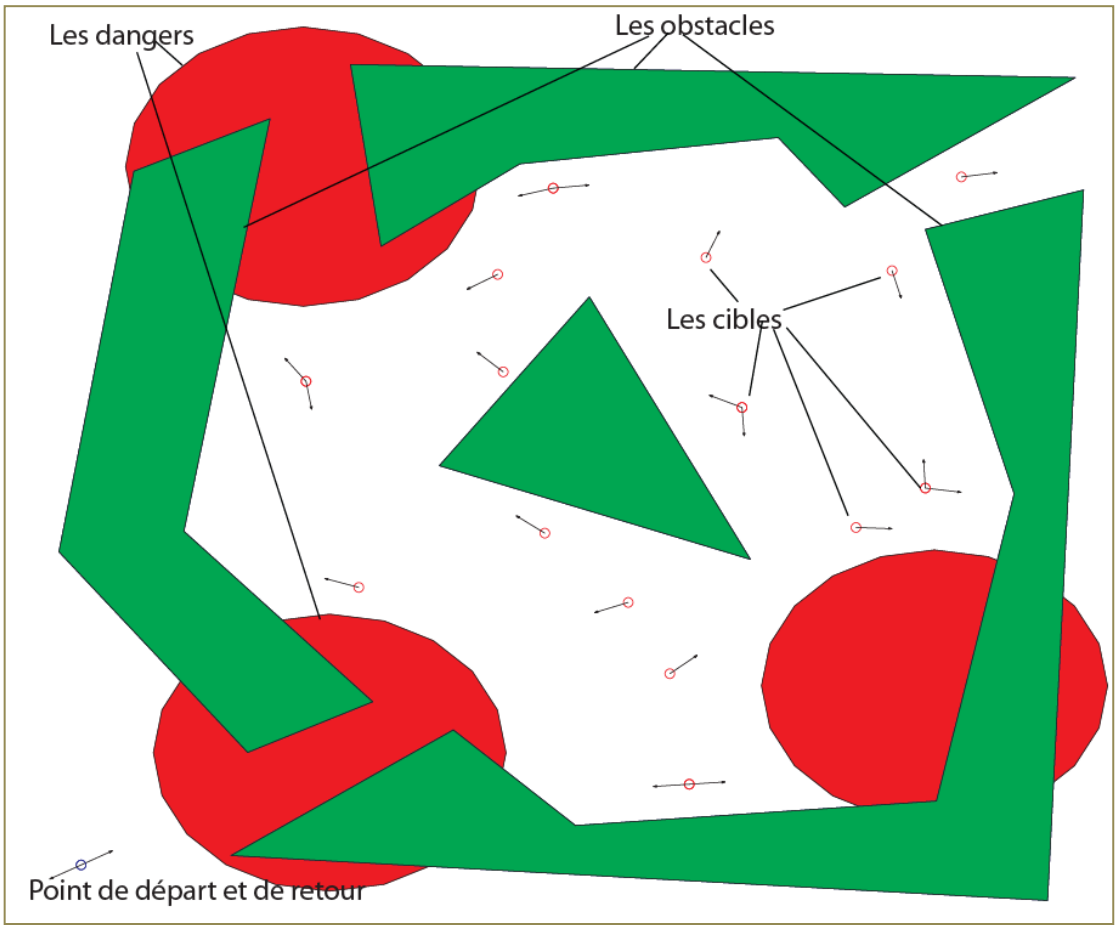


Figure 27 : Exemple de terrain où une mission de surveillance est à planifier

S'il s'agit d'une planification de mission, on a suffisamment de temps pour le calcul des plans de vol car le problème est statique. S'il s'agit d'exécuter une mission, le temps de calcul devient crucial car le problème est de nature dynamique : apparition de nouvelles cibles, cibles pas bien identifiées, disparation d'éléments de la flotte. Cependant même si

notre projet porte sur la planification, nous avons pris soin de faire attention au temps de calcul et aux éléments de solutions pour qu'ils soient valables pour l'exécution d'une mission (voir conclusion).

La Figure 27 illustre un exemple de schématisation d'un terrain (voir chapitres précédents) où une mission de surveillance est à planifier. Les obstacles sont représentés par leur enveloppe polygonale qui peut être convexe ou non. Comme dans les chapitres précédents, les zones de danger sont des radars ennemis et sont représentées par des cercles qui schématisent leur portée. Le point de lancement de la mission est en bas à gauche de la figure et est représenté par deux configurations (seule l'orientation de l'avion diffère entre les deux, celle de départ et celle de retour). Finalement, les cibles sont représentées par des configurations formées par des points géométriques sur le terrain et d'orientations. Ces orientations représentent les angles d'attaques idéaux pour identifier les cibles. À certaines cibles correspondent plusieurs configurations, car, pour les identifier il faut plus qu'une visite et à différents angles.

4.1.1 Objectif du problème

L'objectif général de notre problème consiste, donc, à **assurer la couverture d'un territoire donné par une flotte d'UAVs donnée (nombre d'UAVs fixe) lors d'une mission de surveillance, en visitant des points clés sous des contraintes de conditions géographiques, de danger (présence de radars), d'autonomie et de manœuvrabilité des UAVs.**

Ainsi, l'objectif de notre problème est de minimiser le coût de la mission. Le coût de la mission est relié au cumul du temps total passé par les avions pour réaliser la mission. Les vols se font à altitude constante et à vitesse constante et le coût de la mission est donc proportionnel à la durée cumulative de la mission c.-à-d. la longueur des trajectoires. L'objectif est de minimiser la somme des durées de vol de chaque UAV ou la somme des distances parcourues par chaque UAV.

4.1.2 Contraintes du problème

Les principales contraintes de notre problème sont le terrain avec ses obstacles, les zones de danger, la non-holonomie et l'autonomie des UAVs. La modélisation du terrain en termes de graphe orienté (voir chapitres précédents) inclut déjà de la bonne façon les contraintes du terrain, de ses obstacles, de ses zones de danger et de la non-holonomie des UAVs. Il reste, alors, deux contraintes : l'autonomie du véhicule et le risque total de la mission.

Le risque limite est défini sur toute la mission. La contrainte est que le risque total encouru par tous les avions ne dépasse pas un certain seuil. Donc, si un avion prend un chemin trop risqué, les autres ne pourront pas prendre autant de risque. Le risque est ainsi considéré comme une ressource disponible pour tous les avions et consommée partout sur le terrain dépendamment des zones de passages.

L'autonomie du véhicule peut être définie en durée ou en distance car on suppose la vitesse constante. Chaque UAV a sa propre autonomie. Elle dépend de sa trajectoire et non des cibles visitées. On pourra donc aussi la considérer comme une ressource, mais

cette fois-ci définie pour chaque avion et consommée partout sur le terrain dépendamment des trajectoires suivies.

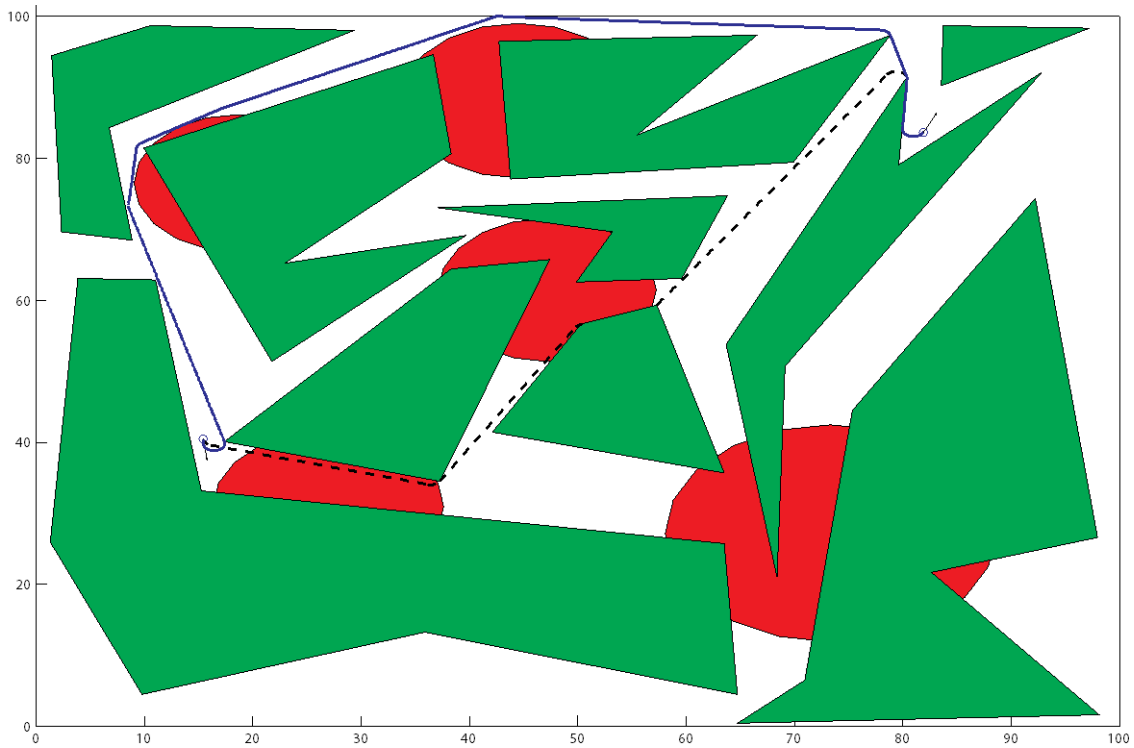


Figure 28 : Exemple de deux trajectoires : une qui consomme plus de risque(en trait pointillé) et une qui consomme plus d'autonomie(en trait plein).

La Figure 28 illustre deux exemples de trajectoires que peut prendre un UAV entre deux mêmes configurations. Une qui consomme plus d'autonomie et aucun risque (celle en trait plein) et une qui consomme plus de risque et moins d'autonomie (celle en trait pointillé). Si l'avion suit la première, il risque de se compromettre et de ne pas finir sa tâche. S'il suit la deuxième il risque de prendre plus de risque et compromettre la mission. La solution idéale pour cet UAV se trouve entre les deux et dépend des autres cibles à visiter et des trajectoires des autres avions de la flotte.

4.1.3 Hypothèses de travail

Les hypothèses de travail sont celles citées dans le chapitre 1, le chapitre de la présentation de la problématique.

4.2 Formulation et modélisation

Le terrain est modélisé sous forme d'un graphe orienté (voir chapitre 3) qui représente la définition du problème de tournées de véhicules. La détermination des routes des avions est formulée sous forme d'un CVRP, problème de tournées de véhicules avec contrainte de capacité. La localisation des cibles à visiter est définie par le graphe qui contient également tous les points intermédiaires nécessaires pour visiter les cibles. Les routes sont des trajectoires non holonomes visitant les cibles affectées au drone en question. Pour chaque avion i nous définissons une ressource en termes d'autonomie ($a_{i,max}$) et pour la mission nous avons une ressource en terme de risque limite (r_{max}). Ces ressources sont consommées sur tout le terrain le long des trajectoires.

Une formulation mathématique du problème est présentée dans ce qui suit. Cette formulation est inspirée des travaux de Dantzig et al. (1954) et adaptée à notre problème. Elle ne sera pas simplifiée car le but est de s'assurer de la bonne formulation du problème et de montrer sa difficulté. Le problème sera résolu par une recherche Tabou.

L'objectif :

$$Min\left[\sum_{UAVi}\left(t_{F_i}\right)\right] \qquad (1)$$

Sous les contraintes :

$$\begin{cases}\left(x_i(0), y_i(0)\right) = point\ de\ lancement; \\ \left(x_i\left(t_{F_i}\right), y_i\left(t_{F_i}\right),\right) = point\ de\ retour;\end{cases} \forall\ i\ un\ UAV \qquad (2)$$

$$\begin{cases} \dot{x}_i(t) = v_i(t)cos\big(\theta_i(t)\big) \\ \dot{y}_i(t) = v_i(t)sin\big(\theta_i(t)\big) \end{cases} \forall\, t \in [0, t_F]\ et\ \forall\, i\ un\ UAV; \qquad (3)$$

$$\cup_{UAV\ i}(route_i) = ensemble\ des\ cibles \qquad (4)$$

$$-\phi_{i,max} \leq \dot{\theta}_i(t) \leq \phi_{i,max}\ \forall\, t \in \big[0, t_{F_i}\big]\ et\ \forall\, i\ un\ UAV; \qquad (5)$$

$$(x_i(t), y_i(t)) \notin \mathcal{O}\ \forall\, t \in \big[0, t_{F_i}\big]\ et\ \forall\, i\ un\ UAV; \qquad (6)$$

$$\sum_{i\ un\ UAV} \left[\int_o^{t_{F_i}} r\big(x_i(t), y_i(t)\big)\, dt\right] \leq r_{max} \qquad (7)$$

$$\int_o^{t_{F_i}} \sqrt{\dot{x}_i(t)^2 + \dot{y}_i(t)^2}\, dt \leq a_{i,max};\ \forall\, i\ un\ UAV \qquad (8)$$

Dans cette formulation l'équation (1) représente la fonction objectif. Elle représente le coût total de la mission, c'est-à-dire la somme des durées des vols. Les deux contraintes (2) tiennent compte des points de lancement et de retour des UAVs. Les contraintes (3) sont des équations différentielles qui relient les commandes de contrôle à la position de l'avion. (4) permet de s'assurer qu'on couvre toutes les cibles où $route_i$.est l'ensemble des cibles affectées au véhicule i. Les contraintes (5) sont les contraintes cinématiques reliées au rayon de courbure minimum représenté ici par $\dot{\theta}(t)$ la variation de la direction de l'avion qui doit être confinée entre $-\phi_{max}$ et ϕ_{max}, l'angle de braquage maximum. (6) signifie que tous les points des trajectoires sont en dehors de la zone des obstacles. (7) définit une limite r_{max} pour le risque pour toute la mission. La dernière contrainte (8) définit $a_{i,max}$, l'autonomie de chaque avion i.

4.3 Résolution : le multi-graphe et la recherche Tabou

Dans cette section nous présentons notre approche de résolution du problème ci-dessus. Cette approche consiste en la modélisation du problème sous forme d'un graphe orienté puis en un multi-graphe. Sur ce dernier on résout un problème de tournées de véhicules

avec double capacité sur les arcs à l'aide de la métaheuristique Tabou qui comporte une double liste de mouvements interdits.

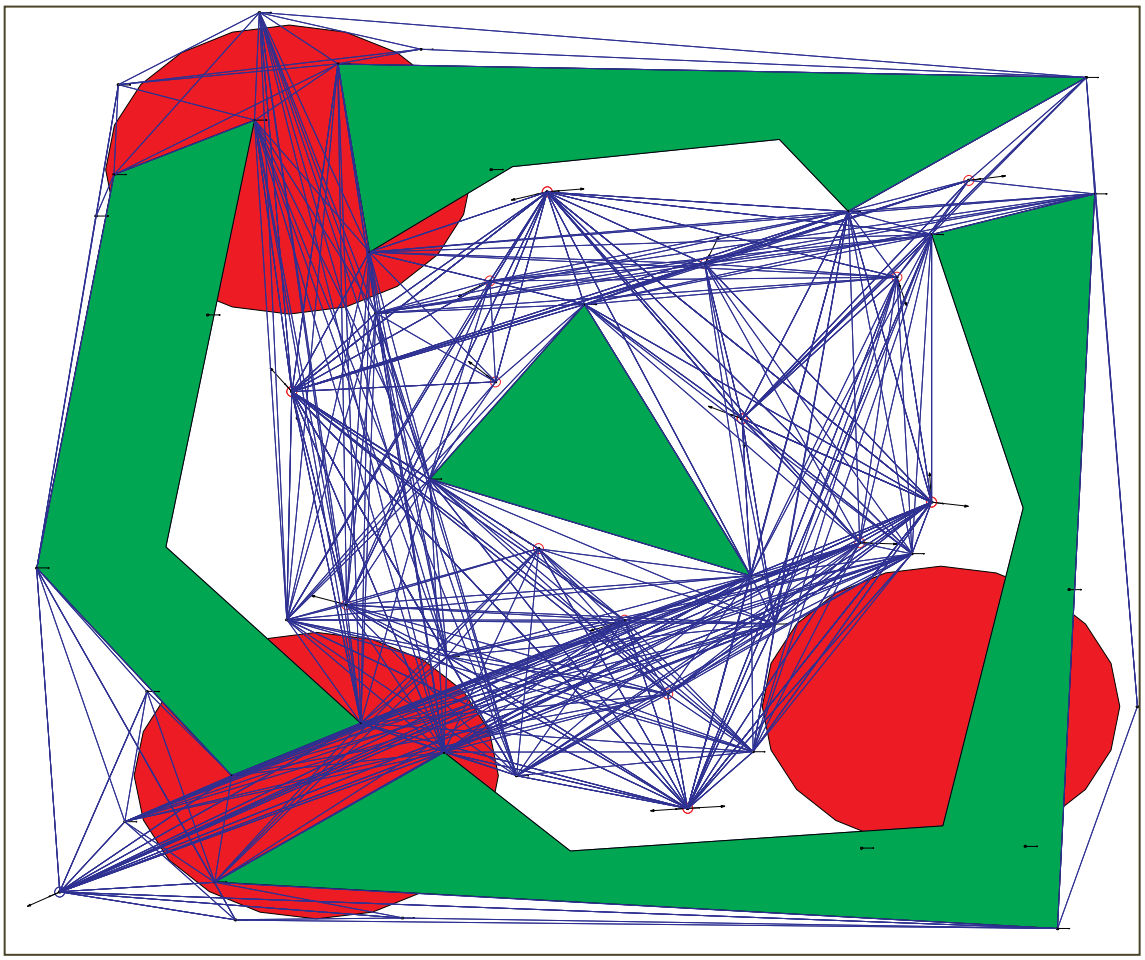


Figure 29 : Graphe orienté correspondant problème non holonome du terrain de la Figure 27.

4.3.1 Modélisation du terrain

Nous avons modélisé le terrain avec toutes ses contraintes sous forme d'un graphe orienté G=(N,A) (voir chapitres précédents). La Figure 29 illustre un exemple de graphe

orienté correspondant au terrain de la Figure 27. Les nœuds de l'ensemble N sont de trois types :

- les deux configurations correspondant aux points de départ et de retour;
- les configurations correspondant aux cibles à visiter (une cible peut correspondre à plusieurs configurations);
- les configurations correspondant aux points intermédiaires nécessaires pour modéliser les obstacles et les zones de danger (voir chapitres précédents).

Chaque configuration (nœud de N) est caractérisée par :

- (x, y) : le couple des coordonnées du point physique lui correspondant sur le terrain;
- θ : la direction, obtenue en discrétisant l'intervalle $[0, 2\pi]$.

4.3.2 La recherche Tabou

La méthode Tabou est une technique de recherche dont les principes ont été proposés pour la première fois par Glover (1986), et elle est devenue très classique en optimisation combinatoire. Elle se distingue des méthodes de recherche locale simples par le recours à un historique des solutions visitées, de façon à rendre la recherche un peu moins « aveugle ». Il devient donc possible de s'extraire d'un minimum local mais pour éviter d'y retomber périodiquement, certaines solutions sont bannies, elles sont rendues « taboues ». Nous avons eu recours à cette métaheuristique à cause du contrôle que nous avons sur le voisinage et à cause de son efficacité prouvée dans les problèmes

d'optimisation combinatoire et surtout sur les grands problèmes classiques, entre autres celui des tournées de véhicules qui nous intéresse ici.

L'idée de la méthode Tabou consiste, à partir d'une solution donnée, à en explorer le voisinage pour trouver une nouvelle solution qui minimise la fonction objectif sur ce voisinage. Durant ce mécanisme, la valeur de la fonction peut augmenter, c'est le cas lorsque tous les points du voisinage ont une valeur plus élevée que la solution courante. C'est, ainsi qu'il est possible d'éviter les extrema locaux. Pour ne pas retomber dans les extrema locaux déjà visités, l'heuristique a recours à de la mémoire, le mécanisme consistant à interdire certains mouvements ou certaines caractéristiques de ces mouvements. C'est ce qu'on appelle la liste taboue, une liste qui contient les mouvements interdits.

Dans cette section nous présentons notre méthode Tabou qui débute par une phase de prétraitement et qui fait intervenir deux listes taboues. Nous présentons d'abord le prétraitement suivi de la définition des ingrédients de la méthode Tabou.

4.3.3 Prétraitement : génération du multi-graphe MG

Dans cette sous-section nous dérivons une phase préalable au Tabou, la génération d'un multi-graphe orienté. On a besoin de ce nouveau graphe car il est plus petit que le graphe orienté de la Figure 29. Il contient beaucoup moins de nœuds car nous ôtons tous les nœuds correspondants aux points intermédiaires qui représentent la majeure partie du graphe orienté. De plus, il ne contient que l'essentiel dont nous avons besoin pour la

méthode Tabou et permet donc de bien traiter les contraintes relatives au risque et à l'autonomie.

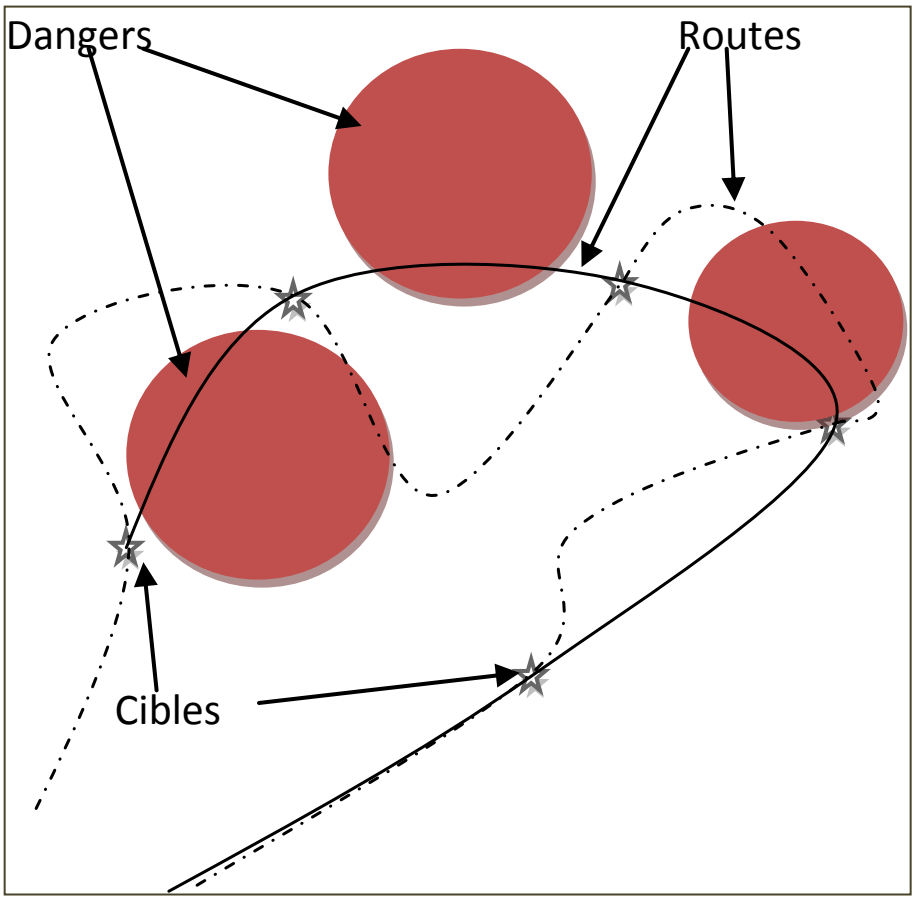


Figure 30 : Exemples de routes visitant les mêmes cibles mais avec des niveaux de risque différents et avec différentes autonomies

Observons tout d'abord que les contraintes de risque et d'autonomie sont en opposition. En général, voir Figure 30, pour visiter les mêmes cibles, un avion peut prendre plusieurs routes qui peuvent être plus ou moins risquées ou de longueur plus ou moins grande. Sur la Figure 30, on voit un exemple de route moins risquée et longue (en trait

pointillé) et une autre plus risquée et courte (en trait plein). Il en existe bien sûr d'autres avec différentes nuances de risque et de longueur.

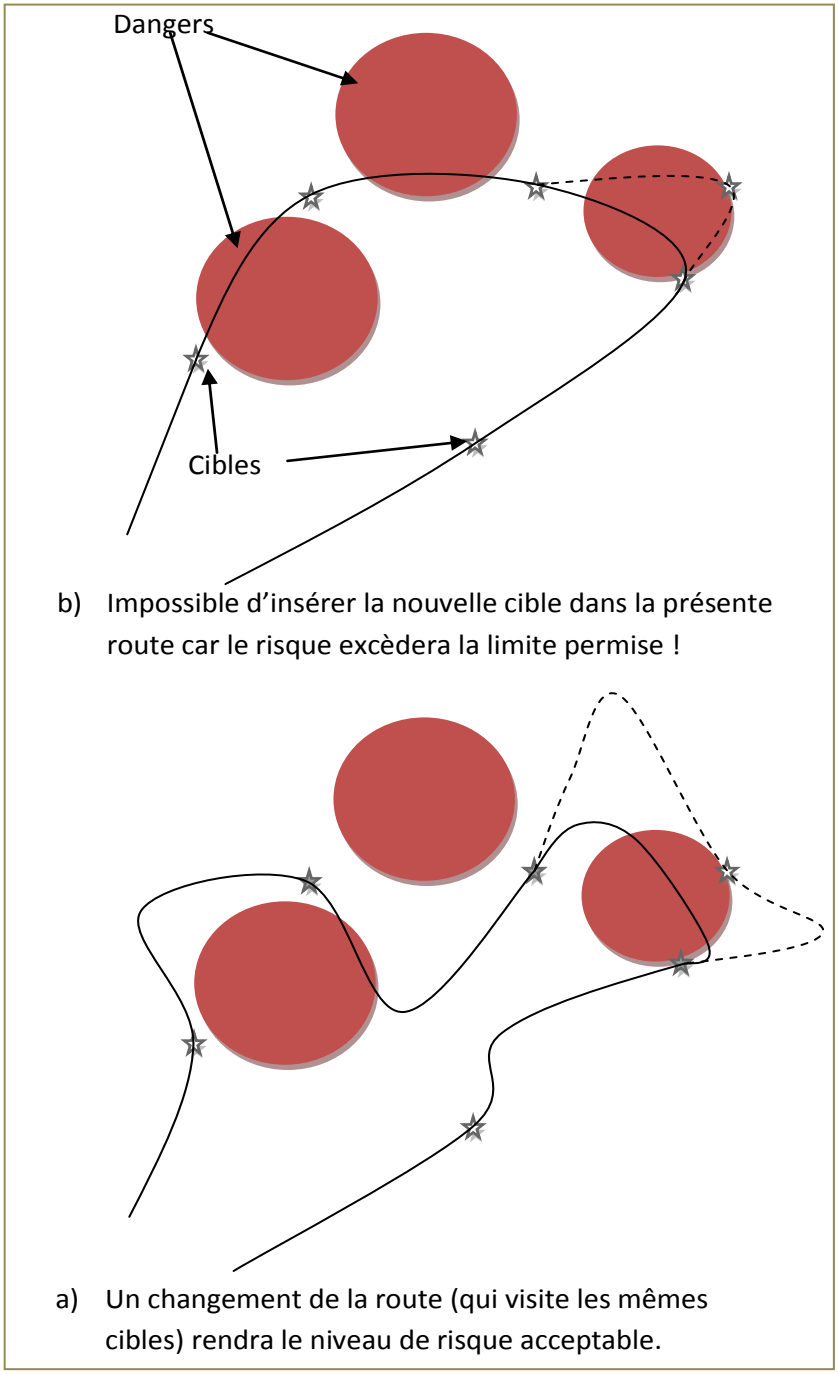


Figure 31 : Exemple de l'implication de l'insertion d'une nouvelle cible dans une route.

Ceci implique que pour visiter un ensemble de cibles, la route peut consommer plus ou moins de risque alloué à la mission et d'autonomie du véhicule. En général, quand la route est risquée, elle est courte, et quand elle est moins risquée, elle est longue. Ce qui pose un problème lors de l'exploration du Tabou, quand on visite le voisinage d'une solution, c'est-à-dire lors de l'insertion d'une cible dans une route.

La Figure 31, illustre ce problème. Dans cette figure, pour insérer la nouvelle cible dans la route existante, il faudrait changer toute la route pour ne pas excéder la limite de risque. Le même problème pourra se présenter du point de vue de l'autonomie. Ceci nécessitera alors, durant l'exploration du voisinage d'une solution, de recalculer toute une route, ce qui augmentera grandement le temps de calcul.

Pour pallier à ce problème nous avons décidé d'évaluer toute les routes possibles, avec différents niveaux de risque et d'autonomie, entre chaque paire de cibles, entre une cible et le point de départ de la mission et entre une cible et le point de retour de la mission. Ce qui implique de doter notre Tabou d'une sorte de mémoire qui stocke toutes les trajectoires intéressantes entre toutes ces paires de nœuds. Ces trajectoires seront qualifiées par leur risque et leur consommation d'autonomie (longueur). Ces informations seront stockées dans un graphe que nous baptisons : le multi-graphe MG.

La génération de ce multi-graphe est faite à partir du travail sur le "Plus court chemin continu non holonome avec risque minimum en présence d'obstacles". Plus précisément, on génère un multi-graphe, MG, totalement connecté où les nœuds sont les cibles et les configurations de lancement et de retour des UAVs. Entre chaque paire de nœuds on a

plusieurs arcs avec des risques et des longueurs différentes (le premier étant le plus long et le moins risqué, le dernier étant le plus court et le plus risqué).

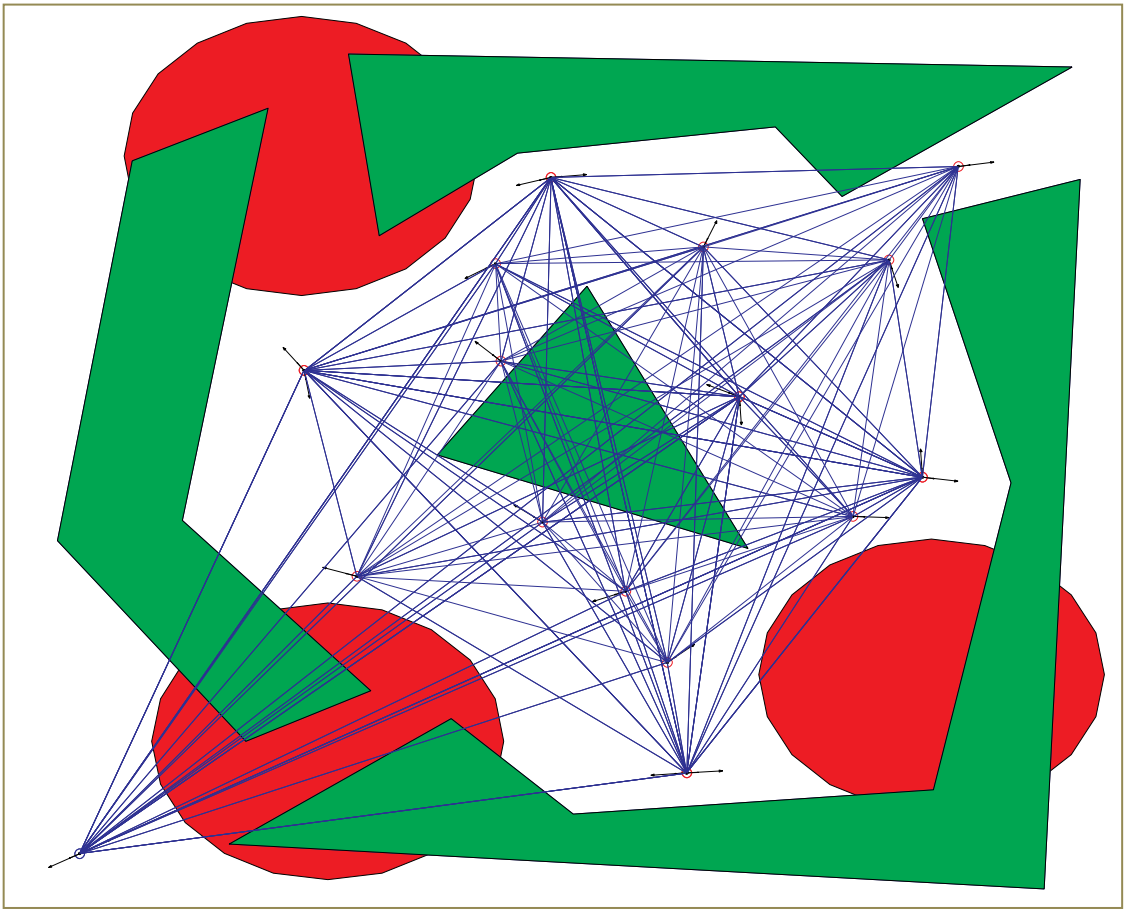


Figure 32 : Le multi-graphe associé au graphe de la Figure 29

La Figure 32 illustre un exemple de multi-graphe associé au graphe de la Figure 29. Les arcs dans la figure ne sont que schématiques et chacun représente plusieurs arcs. Chaque arc correspond à une trajectoire réelle avec un niveau de risque et de longueur donnée. Le Tableau 9, illustre l'exemple d'arcs entre deux nœuds donnés. L'obtention des arcs et des données associés est expliqué et détaillé dans l'algorithme 5.

Tableau 9 : Exemple d'informations stockées dans le multi-graphe entre deux nœuds donnés

Arc	Chemins	Longueur	Risque
Arc 1	[Codage du chemin le moins risqué]	145,30	0,00
Arc 2	[Codage du chemin]	140,35	2,34
Arc 3	[Codage du chemin]	132,68	10,98
Arc 4	[Codage du chemin]	127,09	15,27
Arc 5	[Codage du chemin]	119,52	22,38
Arc 6	[Codage du chemin]	104,82	36,01
Arc 7	[Codage du chemin]	80,25	46,27
Arc 8	[Codage du chemin]	73,28	52,19
Arc 9	[Codage du chemin]	45,30	55,07
Arc 10	[Codage du chemin le plus risqué]	31,32	56,94

4.3.4 L'espace de travail et la représentation des solutions

Pour faciliter la recherche, on relaxe les contraintes de risque et d'autonomie et on considère toutes les solutions, même celles violant ces contraintes. Ainsi, on a plus de possibilités d'exploration. En effet, certaines expériences sur notre problématique ont prouvé que limiter l'espace des solutions aux seules solutions réalisables, limite les performances de notre Tabou.

Chaque solution est composée de r routes (une pour chaque UAV). Chaque route est définie par deux séquences : une donnant les cibles visitées et l'ordre de leur visite, la deuxième donnant l'arc du multi-graphe, utilisé entre chaque paire de cibles. C'est cet arc du multi-graphe qui permet de retrouver dans la mémoire le trajet effectif non holonome du drone.

4.3.5 La fonction d'évaluation

Dans l'espace de solutions visitées, nous tolérons les solutions non réalisables. Ainsi, pour converger vers une bonne solution, nous avons décidé de pénaliser les contraintes violées par les solutions. Les pénalités sont dynamiques et sont incrémentées ou décrémentées tout au long de l'exploration et ce selon que la contrainte en question est violée ou non.

La fonction objectif utilisée par la fonction d'évaluation est donc une fonction qui mesure la durée de la mission additionnée avec des pénalités sur la violation de chacune des contraintes d'autonomie et de risque maximum (voir équation 9). Ces pénalités sont réajustées durant le Tabou.

$$Min\left[\begin{array}{c}\sum_{UAVi}\left(t_{F_i}\right)+p1\times\sum_{i\ un\ UAV}\left[max\left(\int_o^{t_{F_i}}\sqrt{x_i\dot{(t)}^2+y_i\dot{(t)}^2}\,dt-a_{i,max},0\right)\right]\\ +p2\times max\ \left(\sum_{i\ un\ UAV}\left[\int_o^{t_{F_i}}r\left(x_i(t),y_i(t)\right)dt\right]-r_{max},0\right)\end{array}\right] \quad (9)$$

La fonction d'évaluation contient une procédure de correction : à chaque insertion ou changement dans une route, la route entière doit éventuellement être modifiée (voir Figure 31). La procédure de correction a pour tâche de déterminer la nouvelle meilleure route en utilisant le multi-graphe. C'est une procédure de restructuration qui après chaque insertion d'une nouvelle cible, recalcule la meilleure route en réévaluant tous les arcs entre les différentes paires de cibles de la route en question. Plus précisément, lorsqu'une cible est insérée dans une route, nous reparcourrons cette route de son point de départ jusqu'à son point de retour à la base. Pour chaque paire de points consécutifs, nous déterminons l'arc du multi-graphe qui minimise la fonction objectif.

4.3.6 La solution Initiale

Pour créer la solution initiale nous avons utilisé un algorithme d'insertion. À chaque étape, nous insérons la meilleure cible dans la meilleure route et au meilleur emplacement. Cet algorithme est basé sur celui de 'Savings' de Clarke et Wright(1964). Plus précisément, nous démarrons avec des routes vides qui partent du point de départ de la mission et vont directement au point d'arrivée. Puis nous prenons la première route, et nous cherchons la cible qui réalise le plus d'économie en étant insérée dans cette route. On repère cette cible et ce meilleur lieu d'insertion. Si cette insertion ne cause pas un débordement en termes de risque ou d'autonomie, l'insertion est effectuée et nous recommençons. S'il y a débordement on ferme la route et on passe à une deuxième. Et ainsi de suite jusqu'à créer toutes les routes.

Pour décider de la meilleure cible et de la meilleure position d'insertion, on utilise une fonction objectif qui minimise un mélange entre la durée et le risque de la mission. Ce mélange (équation 10) est fixé par un coefficient alpha qui est choisi arbitrairement au départ et, durant l'exploration taboue, on essaye de mémoriser les meilleures valeurs de ce dernier. Ainsi avec cet ingrédient, $\propto$, on peut générer plusieurs solutions initiales.

$$Min\begin{bmatrix} \sum_{i\ un\ UAV}\left[\int_{o}^{t_{F_i}}\sqrt{x_i\dot{(t)}^2+y_i\dot{(t)}^2}\,dt\right] \\ +\propto\times\sum_{i\ un\ UAV}\left[\int_{o}^{t_{F_i}}r\big(x_i(t),y_i(t)\big)\,dt\right]\end{bmatrix} \qquad (10)$$

4.3.7 Le voisinage et les listes taboues

Pour explorer l'espace des solutions, nous définissons deux types de voisinages, un inter-routes et l'autre intra-routes :

- pour le voisinage inter-routes, le mouvement de base consiste à enlever une cible d'une route et l'insérer dans une autre route entre deux autres cibles. Ainsi, chaque solution a un voisinage de taille moyenne $n(r-1)({}^{n}/_{r}+1) \approx \mathcal{O}(n^2)$ (où n est le nombre de cibles à visiter et r le nombre d'UAVs utilisés);
- pour le voisinage intra-routes, le mouvement de base consiste à enlever une cible d'une route et l'insérer dans une autre position de la même route. Ainsi chaque solution a un voisinage de taille moyenne $r({}^{n}/_{r})({}^{n}/_{r}-1) \approx \mathcal{O}({}^{n^2}/_{r})$ (où n est le nombre de cibles à visiter et r le nombre d'UAVs utilisés).

Pour les listes taboues nous en avons défini deux. La première est pour le voisinage inter-routes : nous y stockons, sous forme d'un triplet, le numéro de la cible retirée et les numéros des deux cibles entre lesquelles l'insertion a eu lieu. La deuxième est pour le voisinage intra-routes : nous y stockons, sous forme d'un triplet, le numéro de la route concernée, la cible enlevée et la position où elle a été déplacée. La longueur de chacune de ces deux listes est de l'ordre de la racine carrée de la taille du voisinage :

- une liste taboue inter-routes de taille : $\sqrt{n^2} = n$;

- une liste taboue intra-routes de taille : $\sqrt{n^2/r} = n/\sqrt{r}$.

4.3.8 Le fonctionnement du tabou

Une fois le graphe orienté généré, on commence par la phase de prétraitement où on calcule entre chaque paire de cibles (points de départ et retour inclus) les différents plus courts chemins avec différents seuils de risque pour créer le multi-graphe (voir algorithme 5).

Algorithme 5 : Génération du multi-graphe.

Génération du graphe orienté (modélisation du terrain)
19- Pour chaque paire de cibles faire :
20- Calcul du plus court chemin non holonome sans contrainte sur le risque. (Algorithme de Dijkstra classique)
21- Évaluation du risque de ce chemin => riskMax
22- Pour i=0 jusqu'à p-1 (où p est le nombre d'arcs parallèles qu'on veut avoir dans MG)
23- Calcul du plus court chemin non holonome avec un risque limite de (riskMax x i /n) (voir algorithme 4)

Une fois le multi-graphe créé, on génère une solution initiale, comme décrit dans la section 'solution initiale', et on initialise les listes taboues à vide; On évalue le voisinage inter-routes de la présente solution et on détermine la meilleure solution voisine; Si cette dernière est la meilleure rencontrée jusqu'ici, on la mémorise. L'intensification est faite en explorant le voisinage intra-routes, à chaque fois qu'une meilleure solution est découverte. L'intensification est faite jusqu'à ce que l'objectif ne change pas durant un certain nombre d'itérations (noté IntensificationMax dans le Tableau 11) puis on

retourne à une exploration inter-routes. Si au bout d'un certain temps (un nombre d'itérations noté DiversificationMax dans le Tableau 11) on n'observe aucune amélioration, on déclenche la diversification. Celle-ci se fait en démarrant le Tabou à partir d'une nouvelle solution générée en utilisant un alpha différent (voir la génération d'une solution initiale). Les pénalités quant à elles sont incrémentées quand on viole les contraintes à répétition pendant un certain nombre d'itérations (5 itérations de suite dans les cas testés). Si on les respecte à répétition (pendant 5 itérations de suite dans les cas testés), elles sont décrémentées.

4.4 Tests et résultats

Cette section contient deux groupes de tests : 1) un premier groupe pour valider notre approche, vérifier la qualité de nos solutions et évaluer le comportement du Tabou selon différents paramètres; 2) un deuxième groupe de tests pour évaluer la performance de notre Tabou. Ce dernier est réalisé avec certaines instances de CVRP, celles définies par Christofides et Elion et celles définies par Christofides, Mingozzi et Toth.

Le premier groupe de tests est fait sur trois exemples de terrains que nous avons générés. Nous avons réalisé plusieurs scénarios (quatre) sur chaque terrain. Les dimensions des terrains reflètent la description faite de cas concrets de la défense canadienne. La configuration des terrains, position des cibles, des obstacles et des radars, est faite de telle sorte à être proche de la réalité. Les terrains ont été générés sur Matlab via une interface graphique que nous avons développée et qui permet de dessiner différents types de terrains avec des critères de nombre de cibles, nombre d'obstacles et de zones de

danger. Le Tableau 10 décrit les terrains que nous avons générés. Les schémas des terrains se trouvent à l'annexe 3

Tableau 10 : Description des terrains

Terrains	Nb Obstacles	Nb Dangers	Nb Cibles	Taille du graphe		Taille du multi-graphe	
				Nb Nœuds	Nb Arcs	Nb Nœuds	Nb Arcs
Terrain 0	3	2	10	298	47 085	12	166
Terrain 1	3	2	40	332	67 020	42	861
Terrain 2	4	3	20	399	36 489	22	231

La machine utilisée pour les tests est un ordinateur de type PC dont les caractéristiques sont :

- système d'exploitation : Windows XP;
- processeur : Intel double Core 2 GHz;
- mémoire vive : 4 Go;
- le logiciel : Matlab 2008.a.

4.4.1 Qualité et précisions des solutions

Nous avons testé notre tabou sur les trois terrains, dans un premier temps pour nous assurer que quel que soit le terrain, nous obtenons de vraies solutions pratiques et réalisables. Le résultat est très concluant. En effet, toutes les trajectoires sont non holonomes (voir le zoom sur la Figure 35) et la solution réagit bien et traduit notre tolérance au risque. On le constate bien en comparant les différentes figures (la Figure 33 montre une solution où on tolère un certain niveau de risque et les trajectoires passent donc par les zones de danger; la Figure 34 montre quant à elle une solution où on ne tolère aucun risque).

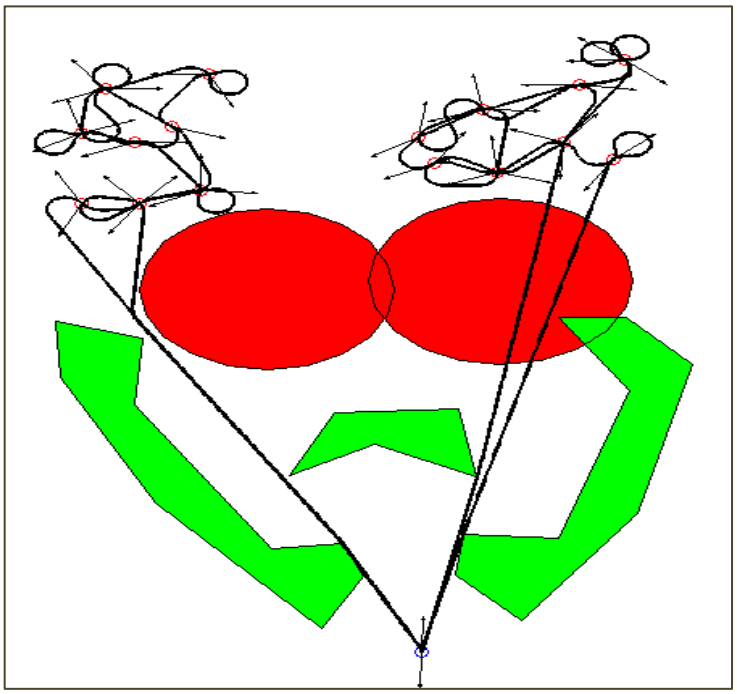


Figure 33 : Terrain 1, solution avec autonomie de 200 et risque max de 15.

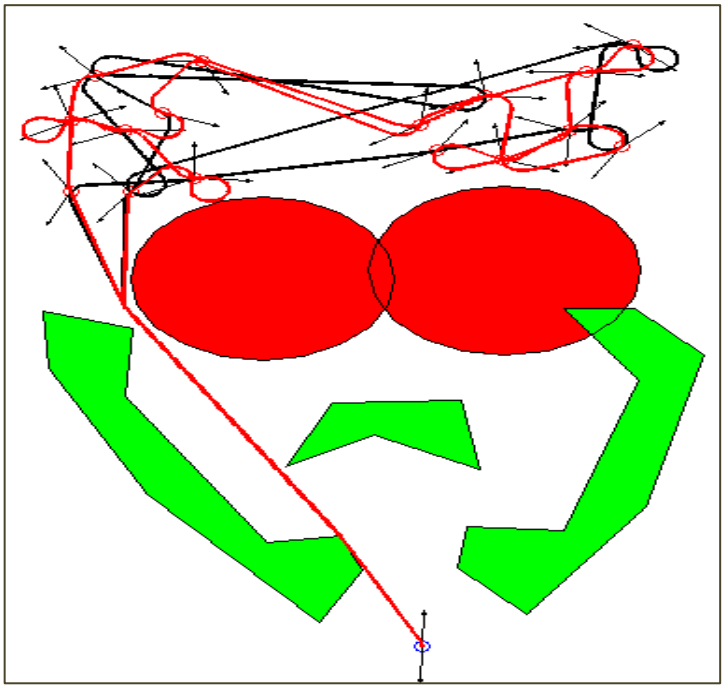


Figure 34 : Terrain 1, solution avec autonomie de 250 et risque max de 0.

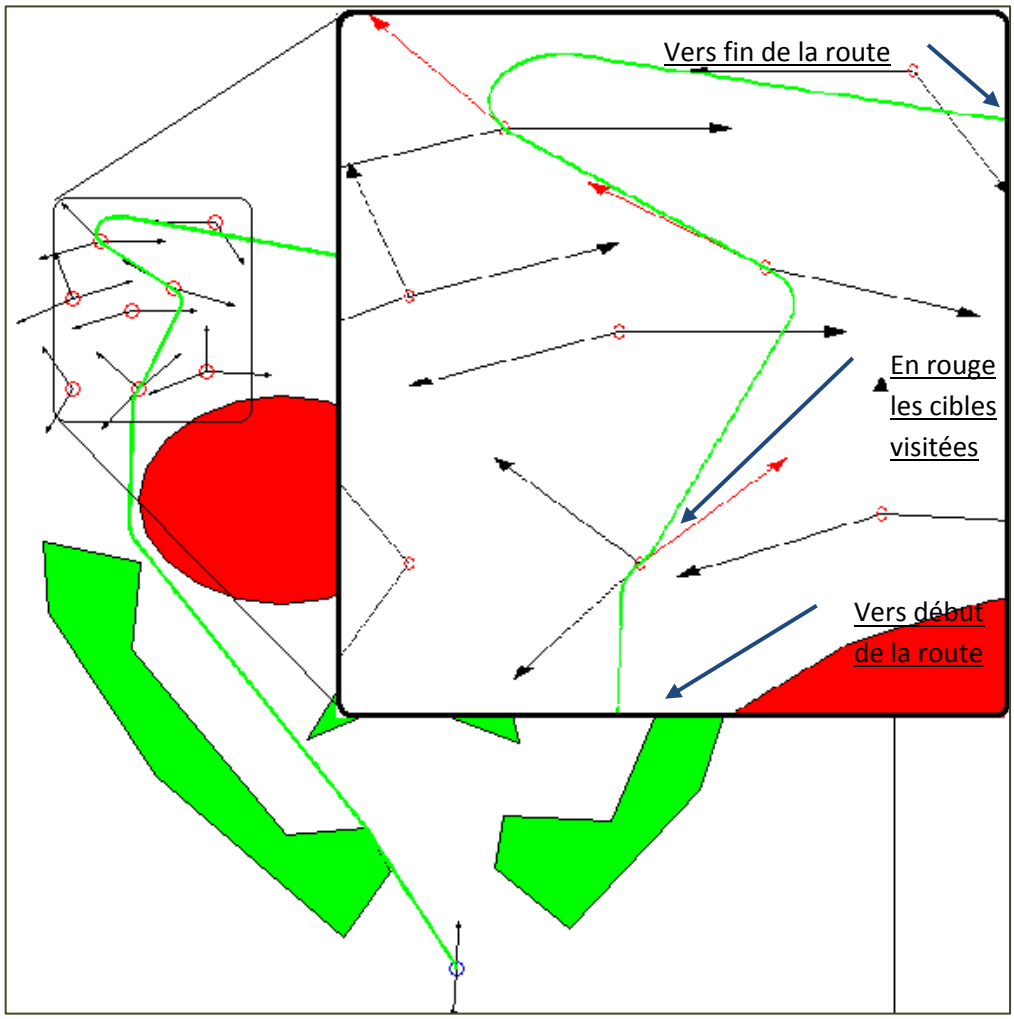


Figure 35 : Détail de la Figure 34 : une partie de la route 1 avec un zoom.

4.4.2 Analyse plus détaillée du comportement du Tabou (exemple Terrain 2)

Sur le terrain 2, nous avons réalisé quatre scénarios avec différentes limites pour les contraintes de risque et d'autonomie. Le Tableau 11 montre les paramètres pour un des scénarios. Dans cette section nous analysons les temps de calcul des solutions, la qualité de la solution et le comportement du Tabou.

Tableau 11 : Paramètres du Tabou pour le terrain 2 scénario 4

% %Problem parameters		
nbUAV = 4;		
Autonomy =	250;	
RiskMax =	300;	
% Pre-processing parameters		
nbSubPath =	10;	% Nb arcs between each two nodes in the Multi-Graph
% Initial solution parameters		
alpha =	1;	% The weight for the risk used in the generation of the initial solution. alpha > 0!
% % The taboo parameters		
maxIterations =	500;	
TabooListInterSize=	20;	
TabooListIntraSize=	10;	
% Intensification		
IntensificationMax=	10;	
IntensificationCount=	0;	
% Diversification		
DiversificationMax=	30;	
DiversificationCount=	0;	
%Penalties		
initialP1 =	1;	% initial penality for automony constrainte
p1 =	initialP1;	
counterP1 =	0;	
CoefP1 =	2;	
counterMaxP1 =	5;	
initialP2 =	0.2;	% initial penality for maximum risk constrainte
p2 =	initialP2;	
counterP2 =	0;	
CoefP2 =	2;	
counterMaxP2 =	5;	

Le Tableau 12 montre les temps de calcul des différents éléments de notre solution pour le cas du terrain 2 avec le scénario 4 : on y voit que la génération du graphe orienté qui modélise le terrain prend beaucoup de temps de calcul, 1210.60 secondes. La grande portion (80 %) de ce temps est due à l'évitement des obstacles non convexes (voir Tableau 3 et section 3.4.1), si ces derniers étaient tous convexes la génération de ce graphe ne coûterait en calcul que 242 secondes. La génération du multi-graphe, 25.47secondes. Ce qui est peu. Dans tous les cas, le calcul du graphe est fait une seule fois et n'influence en rien le Tabou. Cependant, pour le Tabou même, nous constatons

que le temps moyen pour une itération inter-routes est de 0.9564 secondes et celui d'une itération intra-routes est de 0.0743 secondes, ce qui est très rapide, surtout que quelques itérations suffisent à améliorer l'objectif (voir un extrait des itérations à l'Annexe 4).

Tableau 12 : Les temps de calcul pour le terrain 2.

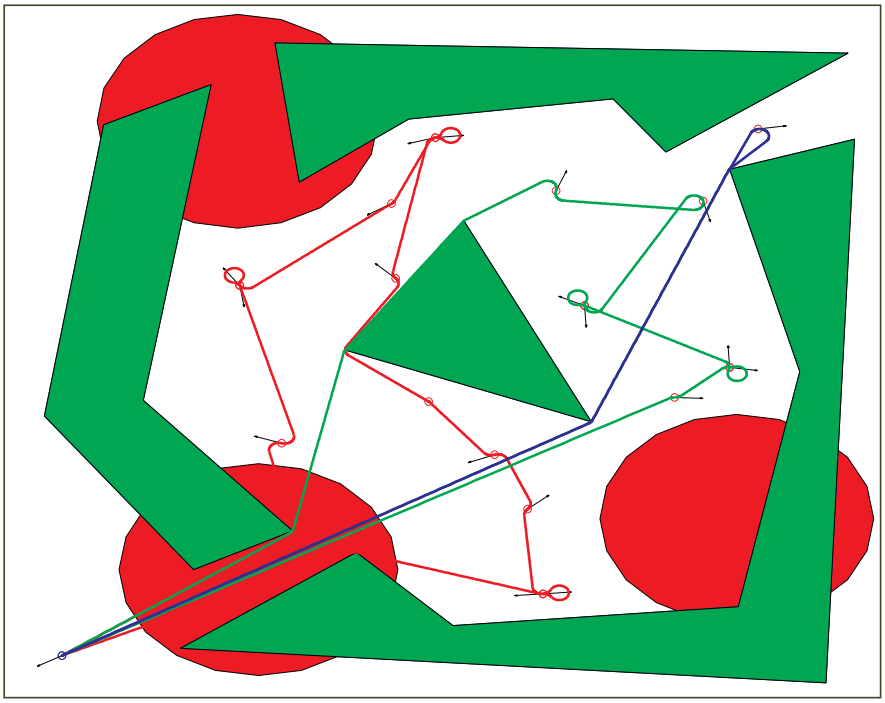
Tps création		Durée d'une itération	
Graphe	Multi-Graphe	Inter-routes	Intra-routes
1210,6 sec	25,47 sec	0,9564 sec	0,0743 sec

Tableau 13 : Amélioration de l'objectif en fonction des itérations et des scénarios pour le terrain 2.

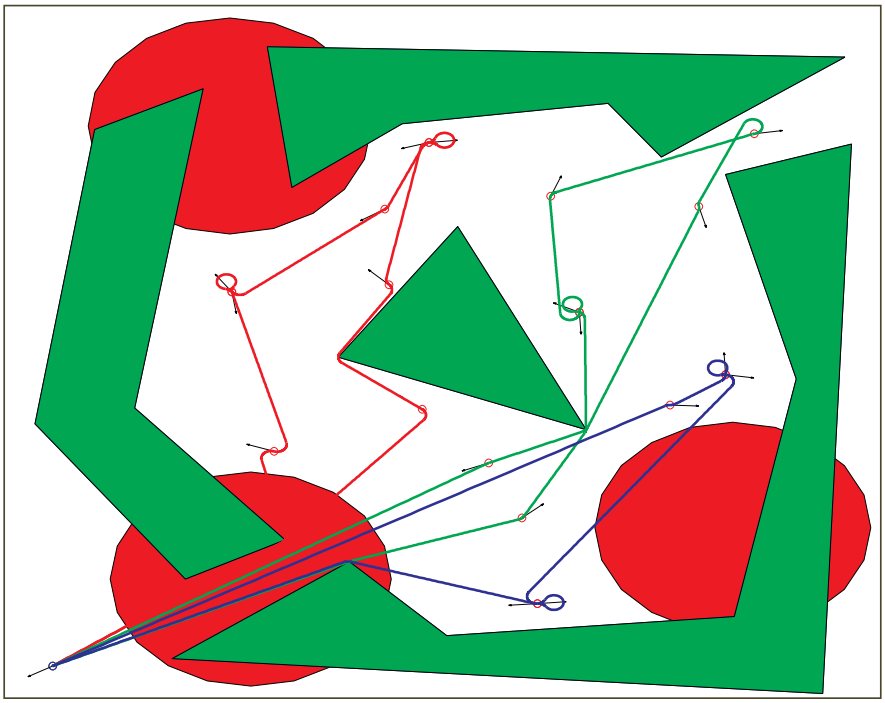
Scénario	**Risque limite**	**L'autonomie**	**Amélioration de l'objectif après l'itération**					
			10	50	100	250	500	2000
1	infini	infinie	19 %	29 %	31 %	41 %	43 %	43 %
2	infini	230	0 %	4 %	9 %	27 %	27 %	27 %
3	50	infinie	0 %	14 %	18 %	18 %	18 %	18 %
4	300	250	0 %	2 %	5 %	10 %	10 %	10 %

Le Tableau 13 montre les performances de notre Tabou. Un bon choix de paramètres a permis de réaliser des améliorations de 10 % à 43 % de l'objectif. Après 500 et jusqu'à 2000 itérations, l'objectif ne change pas pour les 4 scénarios. Le détail des itérations est à l'Annexe 4. La Figure 36 montre les solutions initiale et finale après application du Tabou.

La Figure 37, montre une application du Tabou à partir d'une solution initiale obtenue à l'aide d'un algorithme glouton. Nous avons réalisé une amélioration de 69 % sur cette solution et nous constatons, dans ce cas, que le Tabou a permis aussi de diminuer le nombre d'UAVs, de même autonomie, nécessaires.



a) Solution initiale



b) Solution finale

Figure 36 : Solutions initiale et finale pour le terrain 2 avec scénario 4.

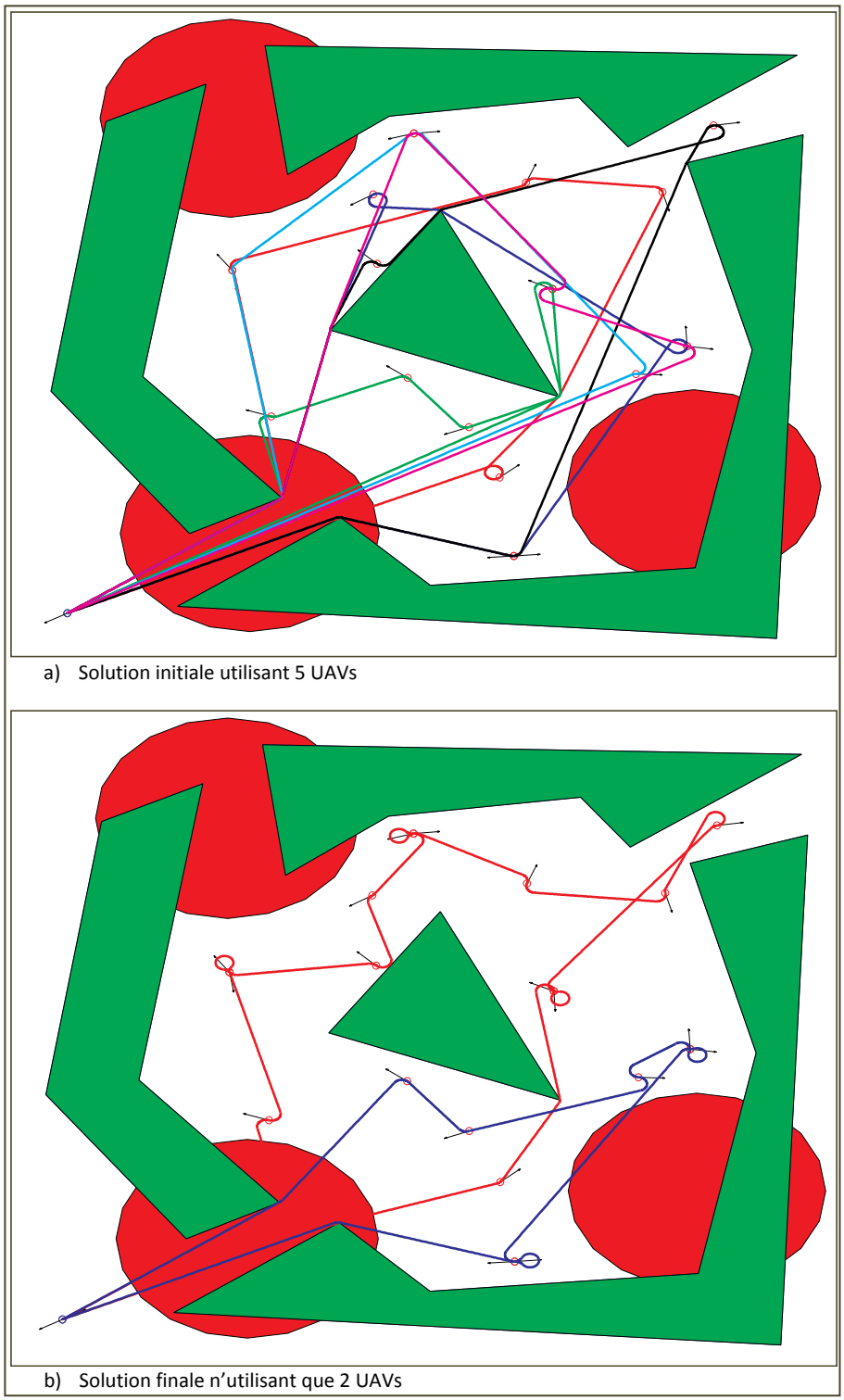


Figure 37 : Exemple de résolution sur le terrain 2, en partant d'une solution (générée par un algorithme glouton).

4.4.3 Calibrage des paramètres Tabou

Dans cette section nous analysons l'impact des changements des paramètres Tabou sur le comportement de ce dernier. Nous détaillons l'analyse de trois paramètres : la taille des deux listes taboues intra et inter route; le degré de précision du multi-graphe, c.-à-d. le nombre d'arcs entre chaque paire de nœuds; et l'impact de la formulation utilisée pour générer le multi-graphe (voir chapitre 3).

4.4.3.1 La taille des listes taboues

Nous avons effectué ces tests sur le terrain 2 avec le scénario 4 (Tableau 11). Nous avons fait varier la taille de chacune des listes et observé les effets sur le comportement du Tabou. Dans un premier lieu, nous fixons la taille de la liste intra-route à 10 et faisons varier celle de la liste inter-route par des pas de 4, de 4 à 40, en regardant l'évolution de la solution qui est résumée dans la Figure 38. Puis en second lieu, nous fixons la liste inter-route à 20 et faisons varier celle de la liste intra-route par des pas de 2, de 2 à 20; la Figure 39 montre ces résultats.

Les deux figures, Figure 38 et Figure 39, montrent respectivement l'évolution de l'objectif selon les itérations pour différentes longueurs de liste taboue inter-routes et intra-routes. Pour les deux cas, nous constatons que la longueur de la liste taboue a un grand impact sur le comportement de ce dernier : une liste trop grande n'est pas bonne, car elle contraint trop l'algorithme et empêche la visite de solutions intéressantes; une liste trop courte n'est pas bonne non plus, car elle ne permet pas d'éviter les minima locaux.

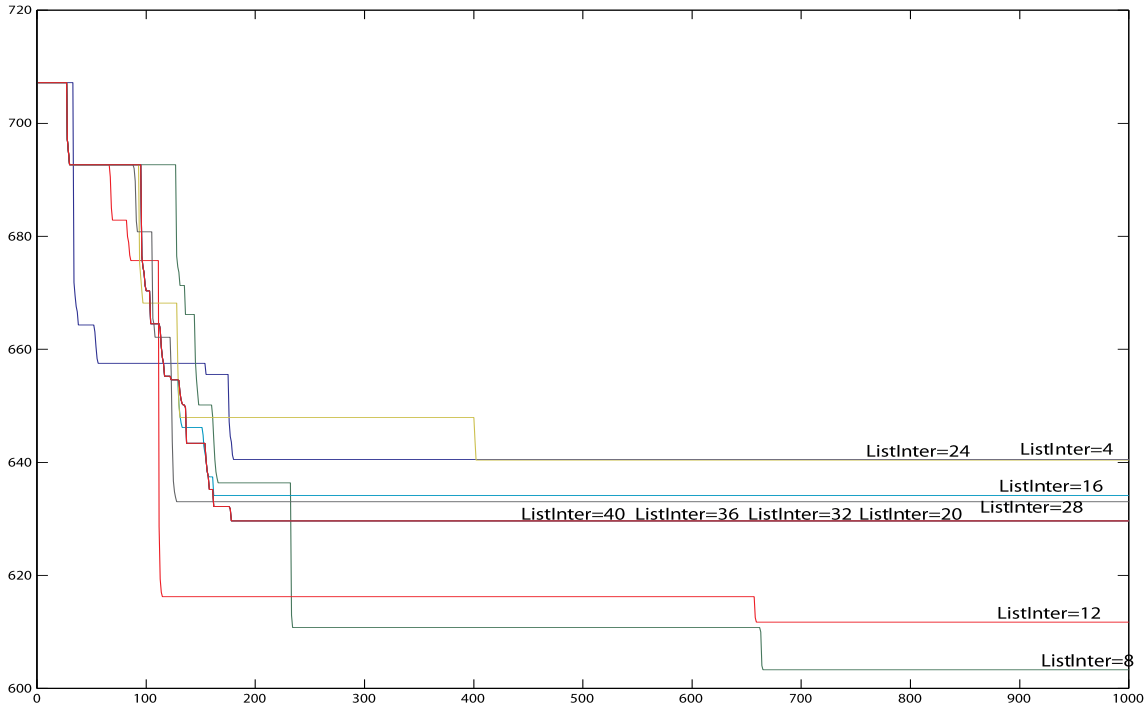


Figure 38 : Évolution de l'objectif en fonction des itérations pour différentes tailles de liste taboue inter-route.

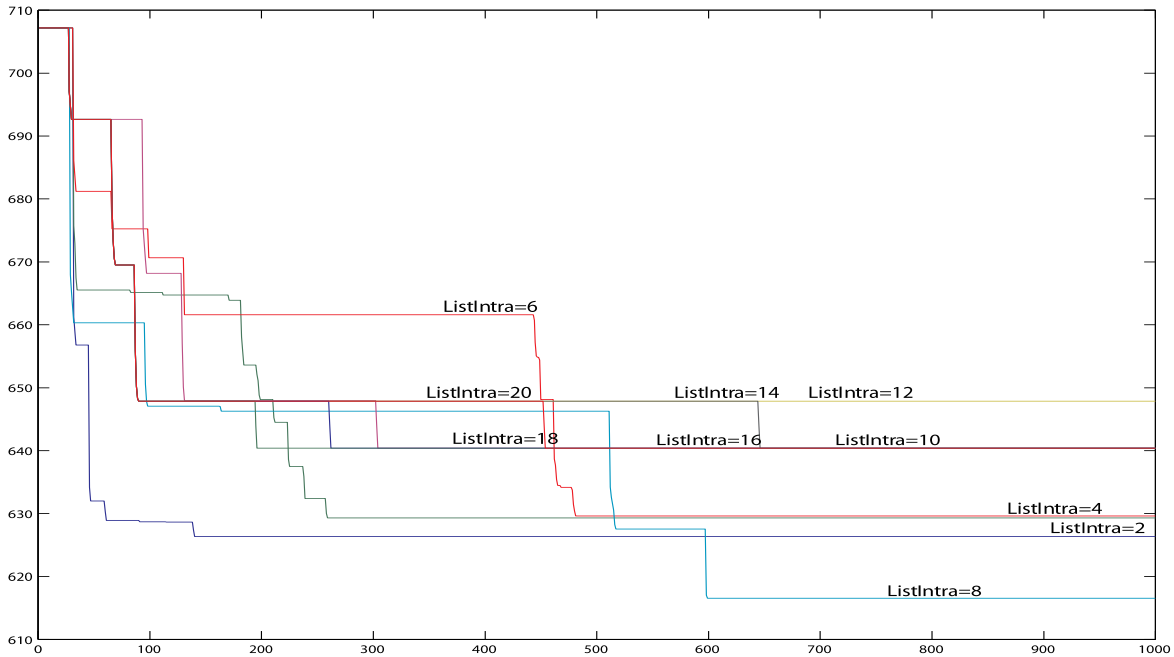


Figure 39 : Évolution de l'objectif en fonction des itérations pour différentes tailles de liste taboue intra-route.

4.4.3.2 Le degré de précision du multi-graphe

Le degré de précision du multi-graphe est défini pas le nombre d'arcs créés entre chaque paire de nœuds. Plus ce nombre est grand plus, a priori, la qualité des résultats serait meilleure, car on aura bien représenté toutes les nuances de risque et d'autonomie possible entre deux configurations données.

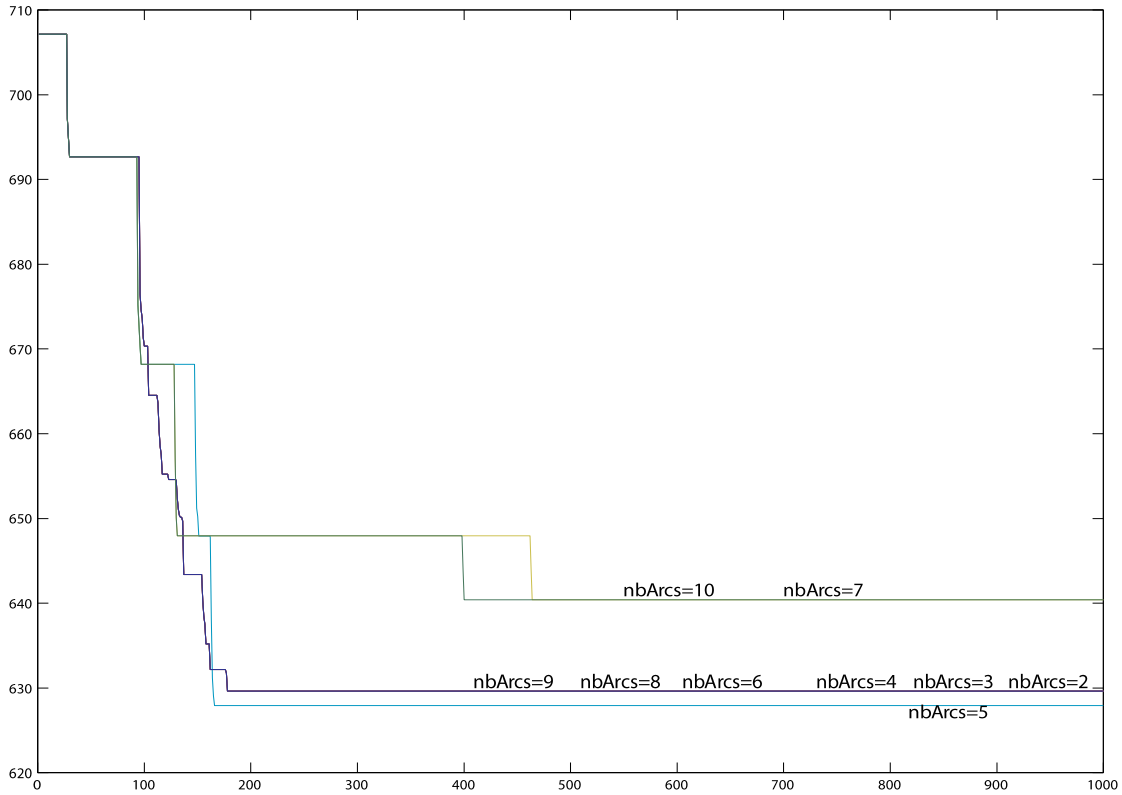


Figure 40 : Évolution de l'objectif en fonction des itérations pour différents degrés de finesse dans le multi-graphe (nombre d'arcs entre chaque paire de nœuds).

Pour vérifier cela, nous avons réalisé des tests sur le terrain 2 avec le scénario 4 (Tableau 11). Nous générons différents multi-graphes avec différents nombres d'arcs (de 2 à 10), utilisons ces multi-graphe pour calculer nos solutions et suivons l'évolution de ces dernières pour différentes itérations. La Figure 40 montre les courbes qui résument

l'expérience. Nous constatons, sur cette figure, que contrairement à ce que nous pensions a priori, ni la solution ni le comportement du Tabou ne sont pas meilleurs quand le graphe est plus fin, c.-à-d. avec plus d'arcs. Ceci peut être expliqué par deux choses : dans la fonction d'évaluation du Tabou, nous avons une procédure de correction des routes (voir section 4.3.5) qui utilise le multi-graphe et plus ce dernier est énorme et plus cette dernière consomme du temps de calcul et corrige mal les routes; la deuxième raison peut être à cause de la nature discontinue de la distribution du risque dans notre cas, ce qui fat qu'au-delà d'un certain nombre d'arcs la précision du multi-graphe ne s'améliore pas.

4.4.3.3 Impact du type de formulation

La définition de notre problématique de planification est faite par une contrainte de risque limite et le multi-graphe est généré en formulant le problème du plus court chemin (chapitre 3) avec une contrainte de risque limite, la formulation 2 dans ce chapitre 3. Et bien sûr dans tous nos tests, nous avons utilisé cette formulation. Maintenant, vu que dans le chapitre 3 nous avons développé une autre formulation du plus court chemin avec le risque dans la fonction objective (la formulation 1), nous nous demandons si cette formulation aurait un impact sur le comportement du Tabou.

Pour répondre à cette question, nous avons changé la phase de prétraitement du Tabou pour pouvoir générer des multi-graphes avec la formulation 1 du chapitre 3. La nouvelle procédure prend en entrée un entier, qui est le nombre d'arcs désirés entre chaque paire de nœuds du multi-graphe et retourne un multi-graphe.

Nous avons réalisé les tests sur le terrain 2 avec le scénario 4 (Tableau 11). Nous générons différents multi-graphes avec différents nombres de d'arcs (de 2 à 10) comme dans la section 4.4.3.2, nous utilisons ces multi-graphe pour calculer nos solutions et suivons l'évolution de ces dernières pour différentes itérations. La Figure 41 synthétise l'expérience.

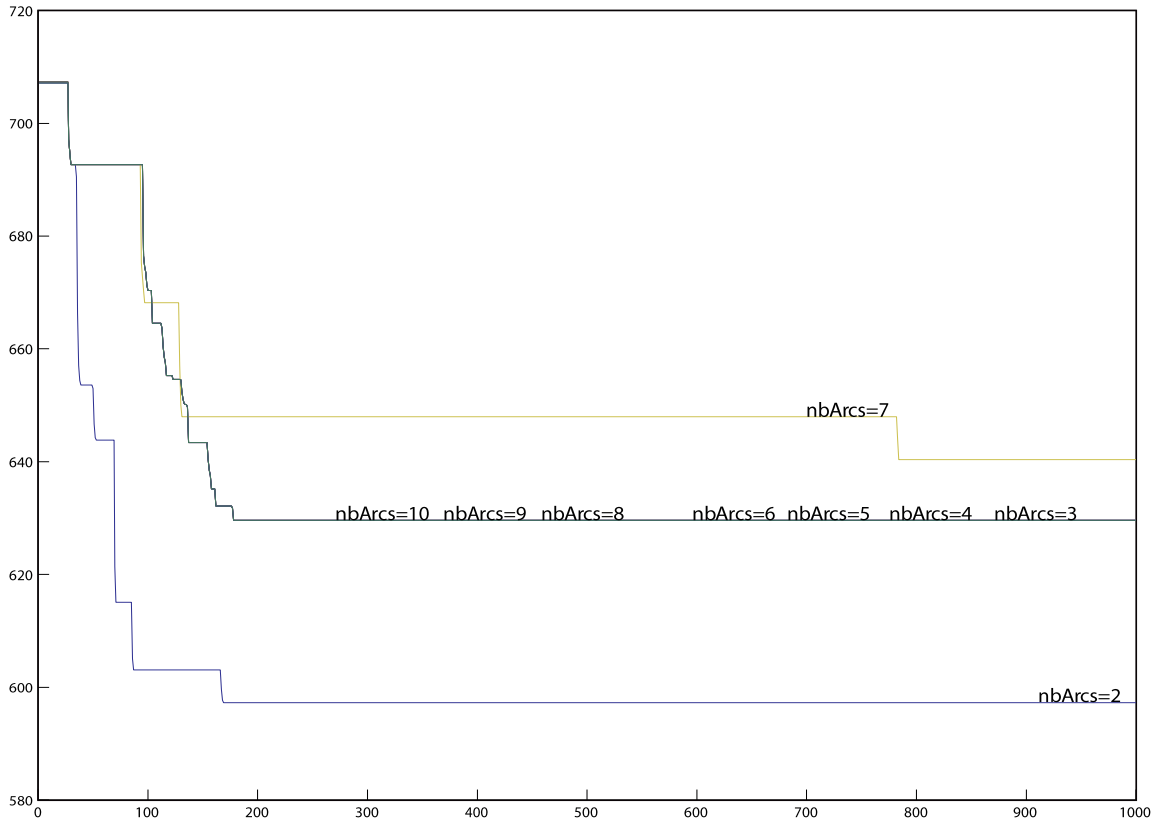


Figure 41 : Évolution de l'objectif en fonction des itérations pour différents degrés de finesse dans le multi-graphe qui est généré selon la formulation 1 (voir chapitre 3).

La Figure 40 et la Figure 41 montrent des résultats pour le même terrain et les mêmes paramètres (Tableau 11) mais avec des multi-graphes générés différemment. La première avec la formulation 2, risque limite dans la contrainte, et la deuxième avec la formulation 1, risque dans l'objectif. À première vue, les deux figures indiquent le

même comportement. Cependant, si nous regardons plus en détail, nous constatons que : 1) pour un nombre d'arcs fixe les deux formulations ne donnent pas le même comportement et aucun lien ne ressort de ces deux figures, c'est-à-dire le nombre d'arcs qui permettent la meilleure solution pour la formulation 1 n'est pas le même que celui pour la formulation 2; 2) dans la formulation 1 et pour un nombre d'arcs égal à 2, nous atteignons un meilleur objectif que par la formulation 2, qui n'a jamais pu permettre le même dans toutes nos expériences. Cela ne nous permettra pas de conclure que la formulation 1 est meilleure car ce n'est qu'une seule expérience. Cependant, nous pouvons affirmer que la formulation a un impact sur le comportement du Tabou.

4.4.4 Évaluation des performances du Tabou sur des instances de CVRP

Ce deuxième groupe de tests, qui a pour but d'évaluer la performance de notre Tabou, est fait sur deux familles d'instances de CVRP : 1) celles définies par Christofides et Elion et qui sont notées E-nx-ky avec x le nombre de nœuds et y la taille de la flotte; 2) celles définies par Christofides, Mingozzi et Toth et qui sont notées M-nx-ky avec x le nombre de nœuds et y la taille de la flotte.

Ces instances de problèmes ne correspondent pas exactement à notre problématique. Nous avons donc eu à les adapter à notre cas. Chaque instance est définie par des nœuds qui ont chacun une demande d_i et une flotte homogène dont les véhicules ont la même capacité c. Dans un premier lieu nous avons associé à chaque instance un terrain où les cibles sont les clients et le dépôt représente les configurations de départ et de retour des

drones. La demande en autonomie de chaque route est reliée à la distance parcourue entre les clients. Cependant il nous fallait définir le risque. Nous l'avons défini comme étant localisé au niveau des clients (les cibles) et correspond à la demande d_i. C'est comme si chaque cible est surveillée par un radar localisé au même endroit que la cible. Et pour associer ce risque aux arcs, nous avons postulé que tout arc arrivant vers la cible i encourt un risque de $d_i/2$ et chaque arc partant de la cible i encourt aussi un risque de $d_i/2$. La limite du risque est la capacité des véhicules. Cependant la limite du risque dans notre problématique est définie sur toute la mission alors que dans ces instances de CVRP, la capacité est définie par véhicule. Alors pour tester notre approche sur ces instances et pouvoir comparer avec les solutions existantes, nous avons défini pour chaque instance 4 scénarios du 1 qui est proche du cas du CVRP au 4 qui est plus du type de notre problématique :

1) Scénario 1 : c'est le scénario le plus proche du cas de l'instance et c'est celui à comparer avec la meilleure solution connue pour l'instance en question. Pour ce scénario nous avons relaxé les contraintes d'autonomie et de non-holonomie et on ne considère aucun obstacle. De plus nous avons changé notre Tabou pour définir la contrainte du risque limite par drone. Ainsi nous sommes aux mêmes conditions que ceux d'un CVRP.

2) Scénario 2 : c'est un scénario intermédiaire. C'est le même que le scénario 1 mais avec la contrainte de risque limite pour toute la mission et non par UAV.

3) Scénario 3 : c'est un scénario intermédiaire. C'est le même que le scénario 2 mais avec la contrainte de non-holonomie en plus. Alors pour définir des configurations au niveau des cibles nous avons associé aléatoirement une direction à chaque cible.

4) Scénario 4 : c'est le scénario correspondant au cas de notre problématique. C'est le même que le scénario 3 mais avec des obstacles que nous avons dessiné le terrain.

Tableau 14 : Résultats obtenus sur les instances de CVRP pour le scénario 1 et au bout de 500 itérations.

Instance	Nb nœuds	Nb UAVs	Solution			Meilleure solution connue	Écart entre solution finale et meilleure solution
			initiale	finale (après 500 iter)	Amélioration		
E-n22-k4	22	4	684.5	**409.4**	40.2 %	**375.0**	**9 %**
E-n23-k3	23	3	683.1	**602.7**	11.8 %	**569.0**	**6 %**
E-n30-k3	30	3	722.8	**674.1**	6.7 %	**534.0**	**26 %**
E-n31-k7	31	7	851.5	**589.0**	30.8 %	**379.0**	**55 %**
E-n33-k4	33	4	1206.3	**935.1**	22.5 %	**835.0**	**12 %**
E-n51-k5	51	5	706.6	**674.0**	4.6 %	**521.0**	**29 %**
E-n76-k7	76	7	956.1	**811.0**	15.2 %	**682.0**	**19 %**
E-n76-k8	76	8	1009.2	**852.9**	15.5 %	**735.0**	**16 %**
E-n76-k10	76	10	1085.3	**983.1**	9.4 %	**830.0**	**18 %**
E-n76-k14	76	14	1353.6	**1353.6**	0.0 %	**1021.0**	**33 %**
E-n101-k8	101	8	1193.2	**1029.2**	13.7 %	**817.0**	**26 %**
E-n101-k14	101	14	1444.0	**1247.6**	13.6 %	**1071.0**	**16 %**
M-n101-k10	101	10	1186.1	**908.5**	23.4 %	**820.0**	**11 %**
M-n121-k7	121	7	1259.9	**1259.9**	0.0 %	**1034.0**	**22 %**
M-n151-k12	151	12	1438.0	**1342.3**	6.7 %	**1053.0**	**27 %**
M-n200-k16	200	16	1828.7	**1510.5**	17.4 %	**-**	**-**
M-n200-k17	200	17	1828.7	**1510.8**	17.4 %	**1373.0**	**10 %**

Dans le Tableau 14, on voit les performances de notre tabou par rapport aux meilleures solutions connues. Nous constatons que les résultats sont assez bons, surtout que c'est seulement au bout de 500 itérations et que le calibrage des paramètres du Tabou n'a pas été fait pour chaque cas séparément. Nous avons utilisé les mêmes paramètres pour toutes les instances sauf pour les tailles des listes taboues qui ont été adaptées selon les formules de la section 4.3.7.

Tableau 15 : Résultats obtenus après 500 itérations pour les différentes instances de CVRP et pour les 4 scénarios. (Les résultats sont exprimés en pourcentage de la meilleure solution connue sauf pour 'M-n200-k16'.).

Instance	Scénario 1		Scénario 2		Scénario 3		Scénario 4	
	Solution initiale	Solution finale	Solution initiale	Solution finale	Solution initiale	Solution finale	Solution initiale	Solution finale
E-n22-k4	183 %	109 %	183 %	76 %	195 %	85 %	220 %	104 %
E-n23-k3	120 %	106 %	120 %	84 %	128 %	90 %	171 %	103 %
E-n30-k3	135 %	126 %	135 %	74 %	145 %	84 %	185 %	99 %
E-n31-k7	225 %	155 %	225 %	109 %	239 %	126 %	310 %	137 %
E-n33-k4	144 %	112 %	144 %	68 %	153 %	77 %	161 %	80 %
E-n51-k5	136 %	129 %	136 %	99 %	146 %	115 %	162 %	128 %
E-n76-k7	140 %	119 %	140 %	100 %	162 %	116 %	210 %	162 %
E-n76-k8	137 %	116 %	137 %	98 %	150 %	111 %	175 %	115 %
E-n76-k10	131 %	118 %	131 %	91 %	156 %	106 %	176 %	126 %
E-n76-k14	133 %	133 %	133 %	74 %	140 %	79 %	161 %	98 %
E-n101-k8	146 %	126 %	146 %	104 %	153 %	124 %	237 %	170 %
E-n101-k14	135 %	116 %	135 %	91 %	155 %	103 %	177 %	116 %
M-n101-k10	145 %	111 %	145 %	87 %	161 %	127 %	213 %	155 %
M-n121-k7	122 %	122 %	122 %	97 %	160 %	116 %	188 %	140 %
M-n151-k12	137 %	127 %	137 %	108 %	167 %	129 %	213 %	170 %
M-n200-k16	(1829)	(1511)	(1829)	(1437)	(2277)	(1940)	(2624)	(2225)
M-n200-k17	133 %	110 %	133 %	113 %	159 %	134 %	180 %	150 %

Le Tableau 15, résume les résultats obtenus sur les différentes instances de CVRP et pour les 4 scénarios utilisés. Dans ce tableau, la solution finale est celle obtenue après

500 itérations. Les résultats sont exprimés en pourcentage de la meilleure solution connue. Nous avons utilisé les mêmes paramètres Tabou sauf pour les longueurs des listes taboues. Si le calibrage de ces paramètres avait été adapté au type de problème, les résultats auraient été meilleurs. Dans le tableau, nous pouvons voir que dans certains cas (seulement ceux des scénarios 2 à 4) la solution est inférieure à 100 % de la meilleure solution connue, cela ne veut pas dire que notre Tabou performe mieux! Non, car les scénarios 2 à 4 sont des situations loin du vrai CVRP et les valeurs inférieures à 100 % sont dues au fait que la contrainte du risque (la capacité) est définie, dans ces scénarios, par mission et non par véhicule.

Tableau 16 : Amélioration de l'objectif au bout de 500 itérations pour chaque instance et chaque scénario.

Instance	Amélioration au bout de 500 itérations			
	Scénario 1	Scénario 2	Scénario 3	Scénario 4
E-n22-k4	40.19 %	58.25 %	56.15 %	52.63 %
E-n23-k3	11.77 %	30.19 %	29.72 %	39.70 %
E-n30-k3	6.74 %	45.53 %	41.69 %	46.36 %
E-n31-k7	30.83 %	51.44 %	47.41 %	55.71 %
E-n33-k4	22.48 %	52.88 %	49.48 %	50.10 %
E-n51-k5	4.61 %	27.16 %	21.28 %	21.11 %
E-n76-k7	15.18 %	28.79 %	28.73 %	22.94 %
E-n76-k8	15.49 %	28.96 %	26.00 %	34.43 %
E-n76-k10	9.42 %	30.59 %	32.00 %	28.39 %
E-n76-k14	0.00 %	44.36 %	43.81 %	39.42 %
E-n101-k8	13.74 %	29.01 %	18.84 %	28.04 %
E-n101-k14	13.60 %	32.39 %	33.91 %	34.35 %
M-n101-k10	23.40 %	39.90 %	21.39 %	27.37 %
M-n121-k7	0.00 %	20.51 %	27.68 %	25.64 %
M-n151-k12	6.66 %	21.13 %	22.95 %	19.94 %
M-n200-k16	17.40 %	21.41 %	14.82 %	15.18 %
M-n200-k17	17.38 %	15.30 %	15.69 %	17.11 %

Pour mieux constater les performances de notre Tabou, il faut regarder l'annexe 5 et le Tableau 16, qui présente les améliorations apportées à l'objectif au bout de 500 itérations. Ces améliorations vont de 14,82 % à 58,25 % pour les scénarios 2 à 4. Alors que pour le scénario 1, il y a deux instances 'E-n76-k14' et 'M-n121-k7' pour lesquelles nous ne réalisons aucune amélioration et pour les autres nous avons une diminution qui va de 4,61 % à 40,19 %.

Dans l'annexe 5, nous avons des exemples de routes non holonomes obtenues par notre Tabou. Nous déduisant de tout cela, que notre Tabou a bien performé, il a pu s'attaquer à des problèmes de grandes tailles et a pu construire des routes de bonne qualité.

4.5 Conclusion

Nous concluons que notre approche apporte de très bons résultats. L'amélioration de l'objectif atteint dans certains cas entre 10 % et 43 % de diminution par rapport à celui d'une solution initiale obtenue par l'algorithme de Clark et Wright (1964). Et sur les instances de CVRP, notre Tabou a obtenu des résultats honorables et a réalisé des améliorations de l'objectif qui ont pu atteindre le 58 % dans certains cas. Nous avons réalisé, aussi, une amélioration de 69 %, mais sur une solution générée par un algorithme glouton. Toutes les solutions calculées sont de très bonne qualité : elles respectent les contraintes de non-holonomie, de l'autonomie et du risque. L'introduction du multi-graphe permet de répondre efficacement aux deux contraintes d'autonomie et de risque et améliore les temps de calcul.

Le Tabou dépend énormément de la taille des deux listes taboues, du degré de précision du multi-graphe et de la formulation utilisée pour générer ce dernier. Les listes taboues doivent avoir la bonne longueur pour éviter le cyclage sans trop contraindre le voisinage. Le nombre d'arcs du multi-graphe doit être aussi bien calibré, un plus grand nombre d'arcs n'est pas la meilleure solution, il peut même être le pire choix. La formulation pour générer le multi-graphe (risque dans l'objectif ou dans la contrainte) influence le comportement du Tabou, le choix du type de formulations dépend du problème qu'on a en main.

La génération du graphe orienté demande beaucoup de temps de calcul, mais ceci n'est fait qu'une fois et au début. Pour la planification des missions, ceci ne pose aucun problème. Notons cependant que pour l'exécution d'une mission (et non sa planification), le temps de traitement devient crucial, et notre approche pourrait donc être problématique. Cependant, la façon dont la génération est faite permet de pallier à ce manque. Il suffit de générer un graphe avant le lancement de la mission (ou utiliser celui de la planification) et de le mettre à jour au cours de la mission. Ceci se fait aisément et demande moins de calcul. Cependant, il est à noté que c'est les obstacles sont convexes le temps de génération du graphe peut diminuer de 80 % (voir chapitre 3) ce qui représente un énorme gain.

Une autre amélioration possible consiste à générer un graphe incomplet, c.-à-d. où la contrainte de non-holonomie est relaxée, puis corriger les trajectoires à la fin. Cette approche demande moins de temps de calcul (voir chapitre précédent) et dans le cas des missions aériennes de la défense, la correction des trajectoires est toujours possible. Ceci

n’est pas forcément le cas pour d’autres types de véhicules ou de mission, plus exactement quand la cinématique et les distances entre les obstacles ne permettent pas la correction.

Finalement, notre approche peut facilement s’étendre à d’autres problématiques, que ce soit pour des phases d’exécution au lieu de planification de missions, ou alors avec d’autres objectifs, comme la minimisation de la taille de la flotte, Notre approche peut aussi être utilisée pour d’autres types de véhicules automatisés. L’adaptation est simple et les outils développés sont facilement généralisables.

CHAPITRE 5 CONCLUSION ET TRAVAUX FUTURS

La présente thèse est consacrée à l'étude du problème de la planification de la mission de surveillance d'une flotte de drones autonomes dans un environnement risqué et parsemé d'obstacles. Cette problématique est importante et d'actualité, car l'utilisation de ces drones est en forte croissance, tant dans le domaine militaire que civil. Ceci est dû au fait que les drones permettent de mener à bien des opérations dangereuses sans risque pour l'humain et à moindre coût. De plus, cette problématique est aussi intéressante en raison des divers défis pratiques qu'elle présente : la modélisation du terrain, l'intégration des obstacles, l'intégration des zones de dangers et la non-holonomie des avions. Finalement, d'un point de vue théorique, la gestion de ces drones est un problème de tournées de véhicules (VRP), car nous cherchons à optimiser une série de routes à être servies par une flotte de drones afin de couvrir un territoire donné et ce, à un moindre coût.

Dans cette thèse, nous avons relevé ces défis en présentant des solutions pratiques basées sur la modélisation de la situation sous forme de graphe, le design de plus courte trajectoire non holonome et la recherche Tabou appliquée à un problème de type VRP avec capacité sur les arcs. Dans ce qui suit, nous résumons les conclusions et les principales contributions de cette thèse, et nous présentons des avenues de recherches reliées à cette problématique.

5.1 Les éléments de conclusion

Le premier volet couvert par cette thèse est celui de la modélisation du terrain sous forme d'un graphe orienté (Chapitre 3). C'est un problème complexe, car il fallait y intégrer toutes les contraintes d'évitement des obstacles, de contournement des zones de dangers et de la non-holonomie des véhicules. Nous avons relevé ce défi en modélisant le terrain sous forme de graphe. Nous avons obtenu un graphe orienté modélisant la totalité des contraintes tout en étant de plus petite taille possible et sans perte de précision au niveau des données – contrairement aux méthodes basées sur le maillage. Cependant, le temps de génération de ce graphe est très long, principalement à cause de la partie d'évitement des obstacles et du calcul des courbes de Dubins. Une version incomplète de ce graphe, ne tenant pas compte de la non-holonomie, donne un graphe encore plus petit et dont le temps de calcul est plus court. Par contre, ce graphe incomplet nécessite une correction de la trajectoire qui n'est pas toujours réalisable. Dans notre cas, la correction est toujours possible, car l'espacement entre les obstacles est assez grand pour la permettre. De plus, le calcul du graphe, dans le cadre de notre planification de la mission, ne se fait qu'une seule fois dans une phase de préparation et n'influence pas le temps de calcul de nos solutions.

Le deuxième volet est le calcul d'un plus court chemin continu non holonome dans un environnement risqué et en présence d'obstacles (Chapitre 3). Nous avons développé une solution basée sur le graphe orienté modélisant le terrain. Cette solution est composée de deux formulations. La première avec le risque dans l'objectif, modélisé comme un problème de plus court chemin classique. La deuxième avec le risque dans la

contrainte, modélisé comme un problème de plus court chemin avec contrainte ressource. Les deux formulations donnent les mêmes résultats mais la première est légèrement plus rapide.

Le dernier volet apporte une réponse à notre problématique principale. Le problème de la planification de la mission de surveillance d'une flotte de drones autonomes dans un environnement risqué et en présence d'obstacles (Chapitre 4). La problématique est difficile mais le défi est relevé. Nous avons modélisé cette dernière comme un problème de tournées de véhicules avec contraintes de capacité sur les arcs. Nous avons appliqué une recherche Tabou avec double liste et les résultats sont de bonne qualité. Nous avons obtenu des trajectoires non holonomes réalisables et le coût des missions est très bas. Cependant, le temps de calcul reste assez long. Ce qui pour une phase de planification ne pose pas de problème mais pour un cas temps réel (c.-à-d. la phase d'exécution de la mission), cela reste insuffisant.

5.2 Les contributions

La première contribution de cette thèse est plus pratique et consiste en le rapprochement entre deux domaines : celui du « contrôle des systèmes » et celui de « la recherche opérationnelle ». Nous avons apporté une réponse pratique à la problématique de planification de mission de drones. Nous avons appliqué des outils de la recherche opérationnelle à des problèmes qui étaient traditionnellement traités comme des problèmes de contrôle. Ceci constitue un apport très significatif.

La deuxième contribution de cette thèse réside dans le travail de modélisation du terrain. Une modélisation qui intègre toutes les contraintes réelles du problème (la non-holonomie, l'autonomie, le risque et les obstacles) dans un graphe orienté et qui minimise au maximum la taille de ce dernier. La modélisation est valide quelque soit le degré de précision recherché, contrairement aux méthodes basées sur le quadrillage.

Cette modélisation nous a permis de mieux répondre aux deux questions de design de plus courte trajectoire non holonome et celle de planification de mission d'une flotte de drones dans un environnement risqué et en présence d'obstacles. Ceci constitue une autre contribution qui prendra la forme de deux articles à soumettre et est la principale contribution de la thèse.

5.3 Futures avenues de recherche

La problématique de gestion d'une flotte d'UAVs est très vaste et touche à plusieurs aspects des deux domaines que constituent le « contrôle de systèmes » et la « recherche opérationnelle ». Nous avons contribué à cette problématique à plusieurs niveaux : la modélisation du terrain, le calcul du plus court chemin non holonome et la planification d'une mission de surveillance. Cependant, il reste encore beaucoup à faire.

Notre travail ouvre la porte à d'autres avenues de recherche. Parmi ces avenues, celles qui devront être explorées dans un avenir prochain sont :

- Le temps de calcul de nos trajectoires est long, ce qui posera un énorme problème pour la phase d'exécution de la mission. Peut-on paralléliser notre Tabou pour accélérer encore plus le temps de calcul?

• Chaque drone contient un système embarqué capable de faire des calculs. La parallélisation permettrait de profiter de cette capacité de calcul partagée entre les UAVs. Ce qui pose une autre problématique. Comment synchroniser les calculs de ces drones?

• Ce point est relié aussi à la question du contrôle de la flotte : que se passe-t-il, par exemple, si le contrôle est coopératif plutôt que central? Lequel est bénéfique pour la réussite de la mission? Sachant que les communications entre avions impliquent des coûts et présentent des risques pour la flotte.

• Un autre point est celui de l'identification des cibles. Chaque passage de l'avion au dessus de la cible ne signifie pas que la cible est à 100 % identifiée. On pourra associer une probabilité à chaque situation et ainsi chercher à maximiser l'espérance de la quantité d'information collectée. Ce qui nécessitera une modélisation stochastique pour mieux tenir compte de l'identification des cibles.

• Finalement, s'il est impossible que la flotte d'UAVs réalise toutes les tâches requises, on pourrait pondérer l'exécution de chaque tâche et rechercher une solution de poids total maximum, ce qui s'approche davantage d'un problème de type CTOP (Capacitated Team Orienteering Problem).

BIBLIOGRAPHIE

Alba, E. et Dorronsoro, B. (2006). Computing nice new best-so-far solutions for capacitated VRP with a cellular genetic algorithm. Information Processing Letters, 98:225‑230.

Amberg, A., Domschke, W. et Voß, S. (2000). Multiple center capacitated arc routing problems: A tabu search algorithm using capacitated trees, European Journal of Operational Research, 124:360‑376.

Archetti, C., Hertz, A. et Speranza, M.G. (2006). A tabu search algorithm for the split delivery vehicle routing problem, Transportation Science, 40:64-73.

Archetti, C., Hertz, A. et Speranza, M.G. (2007). Metaheuristics for the Team Orienteering Problem, Journal of Heuristics 13:49-76.

Baker, B.M. et Ayechew, M.A. (2003). A genetic algorithm for the vehicle routing problem. Computers & Operations Research, 30:787‑800.

Baldacci, R., Hadjiconstantinou, E.A., et Mingozzi, A. (2004). An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. Operations Research, 52(5):723–738.

Beasley, J. et Christofides, N. (1989). An Algorithm for the Resource Constrained Shortest Path Problem. Networks, 19:379–394.

Belenguer, J.M., Benavent, E. et Gòmez-Cabrero, D. (2005). Cutting plane and column generation for the capacitated arc routing problem. ORP3 Meeting, Valencia.

Bell, J.E. et McMullen, P.R. (2004). Ant colony optimization techniques for the vehicle routing problem. Advanced Engineering Informatics, 18:41‑48.

Bellaïche, A., Jean, F., et Risler, J.J. (1998). Geometry of Nonholonomic Systems, Robot Motion Planning and Control, 229, Springer (J.-P. Laumond Ed.).

Bellingham, J. S, (2002). Coordination and Control of UAV Fleets using Mixed-Integer Linear Programming, M.ScA. Aeronautic and Astronomy, MIT.

Bellingham, J., Tillerson, M. , Richards, A. et How, J. (2001). Multi-Task Allocation and Path Planning for Cooperating UAVs, in Second Annual Conference on Cooperative Control and Optimization.

Berger, J. et Barkaoui, M. (2003). A new hybrid genetic algorithm for the capacitated vehicle routing problem. Journal of the Operational Research Society, 54:1254‑1262.

Betancourt, A. S. J. (2009). Consensus-Based and Network Control of Fixed Wing Unmanned Aerial Vehicules. Master Thesis. UMI Order Number: 1472897, University of Texas, San Antonio.

Beullens, P., Cattrysse, D., Muyldermans, L. et Van Oudheusden, D. (2003). A guided local search heuristic for the capacitated arc routing problem, European Journal of Operational Research, 147:629‑643.

Bortoff, S. (2000). Path Planning for UAVs. In American Control Conference, 1:364–368.

Brandão, J. (2004). A tabu search algorithm for the open vehicle routing problem. European Journal of Operational Research, 157:552‑564.

Brandão, J. et Eglese, R.W. (2008). A deterministic tabu search algorithm for the capacitated arc routing problem (CARP), Computers and Operations Research, 35/4:1112‑1126.

Carlton, W.B. (1995). A tabu Search to the General Vehicle Routing Problem. Ph.D. dissertation, University of Texas, Austin TX.

Cassandras, C.G. et Li, W. (2002). A Receding Horizon Approch for Solving Some Cooperative Control Problems. Proc. Conf. Decision and control, 3760-3765

Chandler, P. R. et Pachter, M. (1998). Research Issues in Autonomous Control of Tactical UAVs, in Proceedings of the American Control Conference, (Washington, DC), 394–398.

Chandler, P. R., Pachter, M., Swaroop, D., Fowler, J. M., Howlett, J. K., Rasmussen, S., Schumacher, C. et Nygard, K. (2002). Complexity in UAV Cooperative Control, in Proceedings of the American Control Conference, (Anchorage AK).

Chandler, P.R. (2001). UAV Cooperative Control. Proc. American Control Conf, Arlington, 50-55.

Chitour, Y., Jean, F., et Long, R. (2009). A global steering method for general dynamical nonholonomic systems, 48th IEEE Conference on Decision and Control, Shangai.

Clarke, G. et Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points, Operations Research, 12/4:568-581.

Codenotti, B., Manzini, G., Margara, L. et Resta, G. (1996). Perturbation: An efficient technique for the solution of very large instances of the Euclidean TSP, INFORMS Journal on Computing, 8/2:125-133.

Cordeau, J.F., Desaulniers, G., Desrosiers, J., Solomon, M.M., et Soumis, F. (2002). VRP with time windows. In: Toth, P., and Vigo, D. (eds), The Vehicle Routing Problem, 157–193.

Cordeau, J.F., Laporte, G. et Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operational Research Society, 52:928‑936.

Cordeau, J.F., Laporte, G. et Mercier, A. (2004). Improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows. Journal of Operational Research Society, 55:542‑546.

Cordeau, J.F., Laporte, G., Savelsberg, M. et Vigo, D. (2007). Vehicle Routing. In Transportation: Handbooks in Operations Research and Management Science, Barnhart, C. et Laporte, G., North Holland Publishing, 14:367‑428.

Cordeau, J.-F.,Gendreau, M., et Laporte, G. (1997) A tabu search heuristic for periodic and multi-depot vehicle routing problems. Networks, 30:105–119.

Corobléa, B. (1999) Les opérations de distribution du courrier gouvernemental : analyses et voies d'amélioration. Essai de maîtrise, Faculté des sciences de l'administration, Université Laval, Québec, Canada.

Crainic, T.G. et Gendreau, M. (2002). Cooperative Parallel Tabu Search for Capacitated Network Design. Journal of Heuristics 8/6:601-627.

Crainic, T.G. et Semet, F. (2005). Recherche opérationnelle et transport de marchandises. In Optimisation combinatoire: applications, Paschos, V.T., Hermès Lavoisier, 47‑115.

Dantzig, G.B. et Ramser, R.H. (1959). The Truck Dispatching Problem. Management Science 6:80-91.

Dantzig, G.B., Fulkerson, D.R. et Johnson, S.M. (1954). Solution of a Large-Scale Traveling-Salesman Problem, Operations Research, 2: 393-410.

De Jong, K.A. (1975). An Analysis of the Behavior of a Class of Genetic Adaptative Systems. Ph.D. thesis, University of Michigan, U.S.A.

De Mot, J., Kulkarni, V., Gentry, S., Gavrilets, V., Feron, E. (2002). Coordinated path planning for a UAV cluster. AINS Symposium. Los Angeles, California.

Derigs, U. et Kaiser, R. (2007). Applying the attribute based hill climber heuristic to the vehicle routing problem. European Journal of Operational Research, 177:719‑732.

Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M.M., et Soumis, F. (2002). VRP with pickup and delivery. In: Toth, P., and Vigo, D. (eds), The Vehicle Routing Problem, pages 225–242. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia.

Desaulniers, G., Soumis, F. et Laurent, J.-C. (1995). A Shortest Path Algorithm for a Car-Like Robot in a Polygonal Environment, Les cahiers du Gérad G-95-43, École des Hautes Études Commerciales, Montréal, Canada.

Dijkstra, E. W. (1959) A note on two problems in connexion with graphs, Numerische Mathematik, 1:269-271.

Doerner, K.F., Hartl, R.F. et Lucka, M. (2005). A parallel version of the D-ant algorithm for the vehicle routing problem. In Parallel Numerics, Vajtersic, M., Trobec, R., Zinterhof, P. et Uhl, A. Springer, pp.109‑118.

Drezner, Z. (2003). A new genetic algorithm for the quadratic assignment problem. IN-FORMS Journal on Computing, 15:320–330.

Dror, M. et Trudeau, P. (1989). Savings by split delivery routing. Transportation Science, 23: 141-145.

Dubins, L. E. (1957). On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents, American Journal of Mathematics, 79:497-516.

Eglese, R.W. (1994). Routing winter gritting vehicle, Discrete Applied Mathematics, 48:231‑244.

Fisher, M. L. (1994). Optimal Solution of Vehicle Routing Problems Using Minimum K-trees, Operations Research 42:626-642.

Fisher,M. L. et Jaikumar, R. (1981). A Generalized Assignment Heuristic for Vehicle Routing. Networks, 11:109-124.

Fu, Z., Eglese, R.W. et Li, L. (2005). A new tabu search heuristic for the open vehicle routing problem. Journal of the Operational Research Society, 56:267‑274.

Gambardella, L. M., Taillard, E. D. et Agazzi, G. (1999). MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows, In D. Corne, M. Dorigo and F. Glover, editors, New Ideas in Optimization. McGraw-Hill.

Gary, W.K.J (2000). A Hybrid Jump Search And Tabu Search Metaheuristic For The Unmanned Aerial Vehicle (UAV) Routing Problem.

Gendreau, M., Hertz, A. et Laporte, G. (1992). New insertion and post-optimization procedures for the traveling salesman problem, Operations Research, 40:1086-1094.

Gendreau, M., Hertz, A. et Laporte, G. (1994). A tabu search heuristic for the vehicle routing problem, Management Science, 40/10:1276-1290.

Gendreau, M., Iori, M., Laporte, G. et Martello, S. (2006). A tabu search algorithm for a routing and container loading problem. Transportation Science, 40:342‑350.

Gendreau, M., Laporte, G. et Potvin, J.Y. (2002). Metaheuristics for the Capacitated VRP. The vehicle routing problem, Toth, P. et Vigo, D., Society for Industrial and Applied Mathematics, 129‑154.

Gendreau, M., Laporte, G., Musaraganyi, C. et Taillard, É. (1999). A tabu search heuristic for the heterogeneous fleet vehicle routing problem. Computers & Operations Research, 26:1153‑1173.

Gendreau, M., Potvin, J.Y., Bräysy, O. Hasle, G. et Løkketangen, A. (2008). Metaheuristics for the Vehicle Routing Problem and Its Extensions: A Categorized Bibliography. In The Vehicle Routing Problem: Latest Advances and New Challenges, Golden, B. et al., Springer, 143-170.

Glover, F. (1986). Future Paths for Integer Programming and Links to Artificial Intelligence.Computers and Operations Research, 13:533-549.

Glover, F. et Laguna, M. (1997) Tabu Search. Kluwer, Boston.

Goldberg, D. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley Publishing Company Inc, Ney York.

Hertz, A. (2005a). Les métaheuristiques : quelques conseils pour en faire bon usage, in Gestion de Production et Ressources Humaines : méthodes de planification dans les systèmes productifs, Presses Internationales de Polytechnique, 205-222.

Hertz, A. (2005b). Recent trends in arc routing, in Graph Theory, Combinatorics and Algorithmics : Interdisciplinary Applications, (Eds I. Hartman and M. Golumbic), Kluwer, 215-236.

Hertz, A. et de Werra, D. (1991) The Tabu Search Metaheuristic: How We Used It. Annals of Mathematics and Artificial Intelligence, 1:111-121.

Hertz, A. et Mittaz, M. (2001). A variable neighborhood descent algorithm for the undirected capacitated arc routing problem, Transportation Science, 35/4:425-434.

Hertz, A., Laporte, G. et Mittaz, M. (2000). A tabu search heuristic for the capacitated arc routing problem, Operations Research, 48/1:129-135.

Hirabayashi, R., Nishida, N. et Saruwatari, Y. (1992). Tour construction algorithm for the capacitated arc routing problem, Asia-Pacific Journal of Operational Research, 9:155-175.

Hormann, K. et Agathos, A. (2001). "The point in polygon problem for arbitrary polygons", Computational Geometry, 20:131–144.

Jaszkiewicz, A. et Kominek, P. (2003). Genetic local search with distance preserving recombination operator for a vehicle routing problem. European Journal of Operational Research, 151:352‑364.

Kelly, J. et Xu, J. P. (1996). A Network Flow-Based Tabu Search Heuristic for the Vehicle Routing Problem. Transportation Science, 30:379-393.

Kelly, J. P. et Xu, J. (1999). A set-partitioning-based heuristic for the vehicle routing problem, INFORMS Journal on Computing, 11/2:161-172.

Kim, J. et Hespanha, J. (2003). Discrete Approximations to Continuous Shortest-Path: Application to Minimum-Risk Path Planning for Groups of UAVs. In Proc. of the 42nd Conf. on Decision and Contr.

Kinderwater, G. A. P. et Savelsbergh, M. W. P. (1997). Vehicle Routing: Handling Edge Exchanges. In E. H. L. Aarts and J. K. Lenstra (eds), Local Search in Combinatorial Optimization Wiley, Chichester.

Koo, T. j., Ma, Y., Sastry, S. S. (2001). Nonlinear Control Of A Helicopter Based Unmanned Aerial Vehicle Model, IEEE Transactions on Control Systems.

Koo, T.J., Hoffmann, F., Shim, H., Sinopoli, B. et Sastry, S.S. (1998). Hybrid control of model helicopter. In Proceedings of IFAC Workshop on Motion Control, 285-290, Grenoble, France.

Koo, T.J., Sinopoli, B., Sangiovanni-Vincentelli, A. et Sastry, S. (1999). A formal approach to reactive system design: unmanned aerial vehicle flight management system design example. In IEEE International symposium on Computer-Aided Control System Design, 522–527, Kohala Coast, Hawaii.

Kubiak, M. (2004). Systematic construction of recombination operators for the vehicle routing problem. Foundations of Computing and Decision Sciences, 29/3.

Kumar, M. P., Goyal, S., Kuthirummal, S., Jawahar, C. V. et Narayanan, P. J. (2004). "Discrete contours in multiple views: approximation and recognition", Image and Vision Computing, 22.

Kuwata, Y, (2003). Real-time Trajectory Design for Unmanned Aerial Vehicles using Receding Horizon Control, M.ScA. Aeronautic and Astronomy, MIT.

Labadi, N., Prins, C. et Reghioui, M. (2007). GRASP with path relinking for the capacitated arc routing problem with time windows. Lecture Notes in Computer Science, 4448:722‑731.

Lacomme, P., Prins, C. et Tanguy, A. (2004). First competitive ant colony scheme for the CARP. Lecture Notes in Computer Science, 3172:426‑427.

Laporte, G, (1992a). The Traveling Salesman Problem: An overview of exact and approximate algorithms, European Journal of Operational Research, 59: 231-247.

Laporte, G, (1992b). The Vehicle Routing Problem: An overview of exact and approximate algorithms, European Journal of Operational Research, 59: 345-358.

Laporte, G, (2009). The Traveling Salesman Problem, the Vehicle Routing Problem, and their impact on combinatorial optimization, Les cahiers du GERAD, G-2009-57, École des Hautes Études Commerciales, Montréal, Canada.

Laporte, G. et Osman I. H. (1995). Routing problems : A bibliography, Annals of Operations Research, 61:227-262.

Laporte, G. et Semet, F. (2002). Classical Heuristics for the Capacitated VRP. In The vehicle routing problem, Toth, P. et Vigo, D., Society for Industrial and Applied Mathematics, 109-128.

Latombe, J. C. (1991). Robot Motion Planning. Kluwer Academic.

Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G. et Shmoys, D. B. (1985). The traveling salesman problem. A guided tour of combinatorial optimisation. John Wiley et Sons.

Li, X. et Tian, P. (2006). An ant colony system for the open vehicle routing problem. In ANTS 2006, Lecture Notes in Computer Sciences 4150, Dorigo, M. et al., Springer, 356-363.

Magnanti, T.L. (1981) Combinatorial Optimization and Vehicle Fleet Planning : Perspectives and Prospects, Networks, 11:179-214.

Mak, K.-T. et Morton, A. J. (1993). A modified Lin-Kernighan traveling-salesman heuristic, Operations Research Letters, 13:127-132.

Mazzeo, S. et Loiseau, I. (2004). An ant colony algorithm for the capacitated vehicle routing. Electronic Notes in Discrete Mathematics, 18:181-186.

Mehlhorn, K. (2000). Resource Constrained Shortest Paths, Proc. 8th European Symposium on Algorithms (ESA), Springer LNCS, 326-337.

Mester, D., Bräysy, O. et Dullaert, W. (2007). A multi-parametric evolution strategies algorithm for vehicle routing problems. Expert Systems with Applications, 32:508-517.

Moscato, P., et Cotta, C. (2003). A gentle introduction to memetic algorithms. In: Glover, F., and Kochenberger, G.A. (eds), Handbook of Metaheuristics, pages 105–144. Kluwer, Boston.

Naddef, D. et Rinaldi, G. (2002). Branch-and-Cut Algorithms for the Capacitated VRP. In The vehicle routing problem, Toth, P. et Vigo, D., Society for Industrial and Applied Mathematics, 53-84.

Naghi, A. et Soumis, F. (2000). Agrégation des contraintes de ressources en chaque nœud dans un problème de plus court chemin, Les cahiers du Gérad G-2000-47, École des Hautes Études Commerciales, Montréal, Canada.

Nigam, N. (2009) Control and Design of Multiple Unmanned Air Vehicles for Persistent Surveillance. Doctoral Thesis. UMI Order Number: AAI3382798., Stanford University.

O'Rourke, K.P. (1999). Dynamic Unmanned Aerial Vehicle (UAV) Routing With a Java-Encoded Reactive Tabu Search Metaheuristic. MS Thesis AFIT/GOA/ENS/99M. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH.

Okano, H., Misono, S. et Iwano, K. (1999). New TSP Construction Heuristics and Their Relationships to the 2-Opt, Journal of Heuristics, 5:71-88.

Osman I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem, Annals of Operations Research, 41:421-451.

Park, N., Okano, H. et Imai, H. (2000). A Path-Exchange-Type Local Search Algorithm for Vehicle Routing and its Efficient Search Strategy, Journal of ORSJ, 43:197-208.

Parker, L.E. (2000). Current state of the art in distributed autonomous mobile robots, Distributed Autonomous Robotic Systems, Springer, 4:3-12.

Paw, Y. C. (2009). Synthesis and Validation of Flight Control for UAV. Doctoral Thesis. UMI Order Number: 3389355, UNIVERSITY OF MINNESOTA.

Pedersen, M. B., Crainic, T. G., et Madsen, O. B. (2009). Models and Tabu Search Metaheuristics for Service Network Design with Asset-Balance Requirements. Transportation Science, 43/2:158-177.

Pelosi, M. J. (2010). Range Limited UAV Trajectory using Terrain Masking under Radar Detection Risk. Doctoral Thesis. UMI Order Number: 3396712, Nova Southeastern University, Fort Lauderdale, Florida.

Pereira, G. A., Kumar, V., et Campos, M. F. (2008). Closed loop motion planning of cooperating mobile robots using graph connectivity, Robotics and Autonomous Systems, 56/4:373-384.

Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research, 31:1985‑2002.

Psaraftis, H. N. (1983). k-interchange procedures for local search in a precedence-constrained routing problem, European Journal of Operational Research, 13:391-402.

Reeds,J. A. et Shepp, L. A. (1990). Optimal paths for a car that goes both forwards and backwards, Pacific Journal of Mathematics, 145/2:367-393.

Reeves, F. (2003). Genetic algorithms. In: Glover, F., et Kochenberger, G.A. (eds), Handbook of Metaheuristics, pages 55–82. Kluwer, Boston.

Reimann, M., Doerner, K. et Hartl, R.F. (2004). D-ants: Savings based ants divide and conquer the vehicle routing problem. Computers & Operations Research, 31:563‑591.

Renaud, J. et Boctor, F.F. (2002). A Sweep-Based Algorithm for the Fleet Size and Mix Vehicle Routing Problem, European Journal of Operational Research, 140:618-628, Elsevier.

Renaud, J., Boctor, F. F. et Laporte, G. (1996a). A fast composite heuristic for the symmetric traveling salesman problem, INFORMS Journal on Computing, 3/2:134-143.

Renaud, J., Boctor, F. F. et Laporte, G. (1996b). An improved petal heuristic for the vehicle routing problem, Journal of the Operational Research Society, 47:329-336.

Renaud, J., Boctor, F. F. et Ouenniche, J. (2000). A heuristic for the pickup and delivery traveling salesman problem, Computers and Operations Research, 27/9:905-916.

Richards, A., Bellingham, J., Tillerson, M. et How, J. (2002). Coordination and control of multiple UAVs, In AIAA Guidance, Navigation, and Control Conference.

Rochat, Y., et Taillard, E.D. (1995). Probabilistic diversification and intensification in local search for vehicle routing. Journal of Heuristics, 1:147–167.

Ryan, D. M., Hjorring, C. et Glover, F. (1993). Extensions of the Petal Method for Vehicle Routing. Journal of the Operational Research Society, 44:289-296.

Ryan, J.L., Bailey, T.G., Moore, J.T. et Carlton W.B. (1999). Unmanned Aerial Vehicles (UAV) Route Selection Using Reactive Tabu Search, Military Operations Research, 4/3:5-24.

Savelsbergh, M. W. P. (1990). An efficient implementation of local search algorithms for constrained routing problems, European Journal of Operational Research, 47:75-85.

Savelsbergh, M. W. P. et Sol, M., (1995). The general pickup and delivery problem, Transportation Science, 29/1:17-29.

Schoenwald, D.A. (2000). UAVs: In space, air, water and on the ground, IEEE Control Systems Magazine.

Schumacher, C., Chandler, P. R., et Rasmussen, S. (2002). Task Allocation for Wide Area Search Munitions via Network Flow Optimization, in Proceedings of the American Control Conference, (Anchorage AK), 1917–1922.

Shaw, P. (1998). Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems, Proceedings of the Fourth International Conference on Principles and Practice of Constraint Programming (CP '98), M. Maher and J.-F. Puget (eds.), Springer-Verlag, 417-431.

Sigurd, K. et How, J. (2003). UAV Trajectory Design Using Total Field Collision Avoidance, Proceeding of the AIAA Guidance, Navigation, and Control, paper 2003-5728.

Sisson, M.R. (1997). Applying Tabu Heuristic to Wind Influenced, Minimum Risk and Maximum Expected Coverage Routes. MS thesis AFIT/GOA/ENS/97M. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH.

Taillard, É. D. (1993). Parallel Iterative Search Methods for Vehicle Routing Problems. Networks 23:661-673.

Taillard, E., Laporte, G. et Gendreau, M. (1996). Vehicle routeing with multiple use of vehicles. Journal of the Operational Research Society, 47:1065-1070.

Tarantilis, C.D. (2005). Solving the vehicle routing problem with adaptive memory programming methodology. Computers & Operations Research, 32:2309-2327.

Tarantilis, C.D. et Kiranoudis, C.T. (2002). Boneroute: an adaptive memory-based method for effective fleet management. Annals of Operations Research, 115:227-241.

Thompson, P. M. et Psaraftis, H. N. (1993). Cyclic Transfer Algorithms for the Multivehicle Routing and Scheduling Problems, Operations Research, 41:935-946.

Toth, P. et Vigo, D. (2002). The vehicle routing problem, Society for Industrial and Applied Mathematics.

Toth, P. et Vigo, D. (2003). The granular tabu search and its application to the vehicle routing problem. INFORMS Journal on Computing, 15:333-348.

Tsubakitani, S. et Evans, J.R. (1998). An empirical study of a new metaheuristic for the traveling salesman problem, European Journal of Operational Research, 104:113-128.

Van Breedam, A. (1994). An Analysis of the Behavior of Heuristics for the Vehicle Routing Problem for a Selection of Problems with Vehicle-Related. Customer-Related, and Time-Related Constraints. Ph.D. dissertation, University of Antwerp.

Wassan, N.A. (2006). A reactive tabu search for the vehicle routing problem. Journal of the Operational Research Society, 57:111-116.

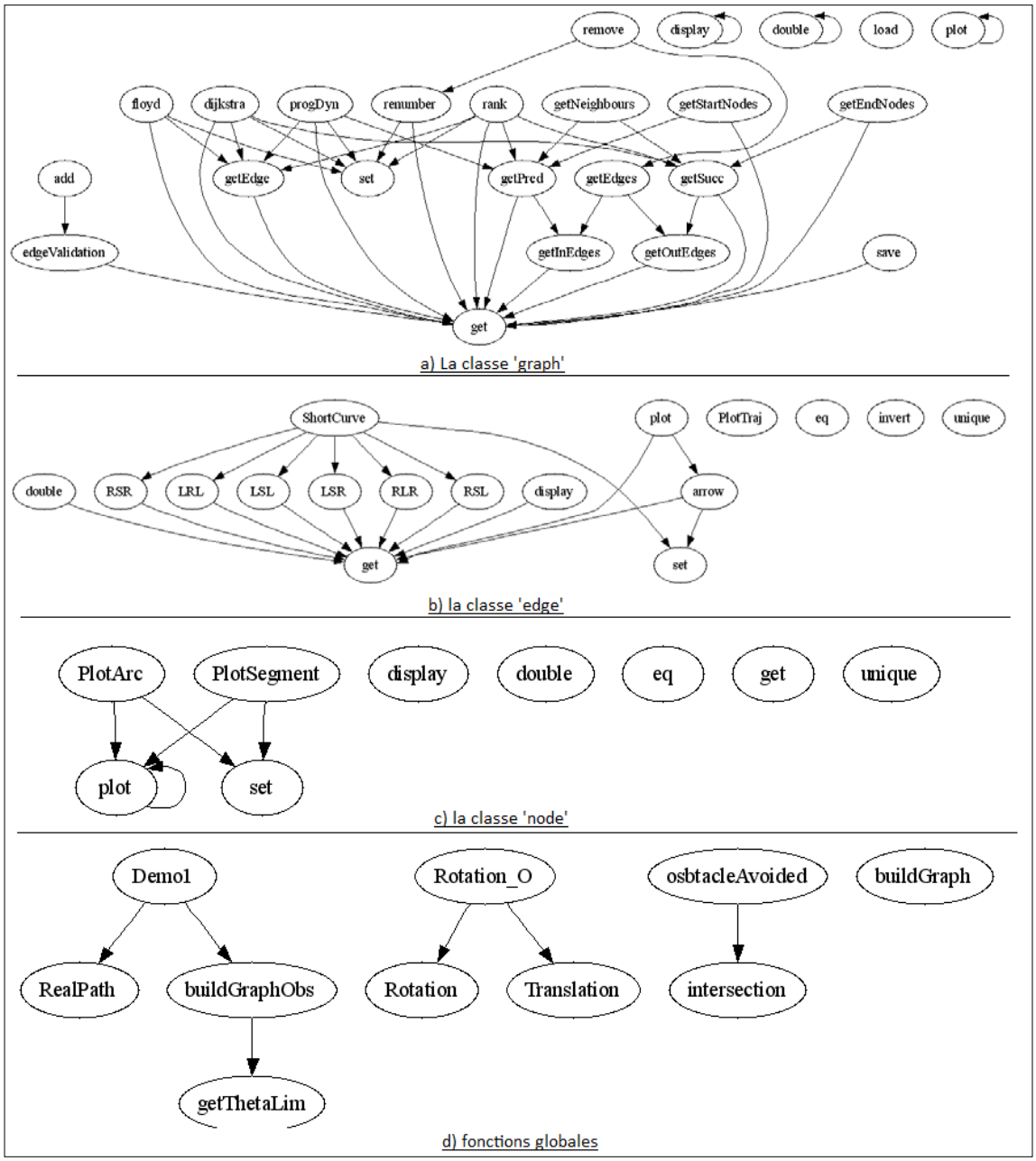
Wøhlk, S. (2005). Contributions to arc routing. Doctoral Thesis, University of Southern, Denmark.

Wøhlk, S. (2008). A Decade of Capacitated Arc Routing. In The Vehicle Routing Problem: Latest Advances and New Challenges, Golden, B. et al., Springer, 29‑48.

Xidias, E. k. et Aspragathos, N. a. (2008). Motion planning for multiple non-holonomic robots: A geometric approach. Robotica 26/4:525-536.

Zeng, I., Ong, H.L. et Ong, K.M. (2005). An assignment-based local search method for solving vehicle routing problems. Asia-Pacific Journal of Operational Research, 22:85‑104.

ANNEXE 1 – L'implémentation de la modélisation du terrain



ANNEXE 2 – Les terrains utilisés pour le plus court chemin

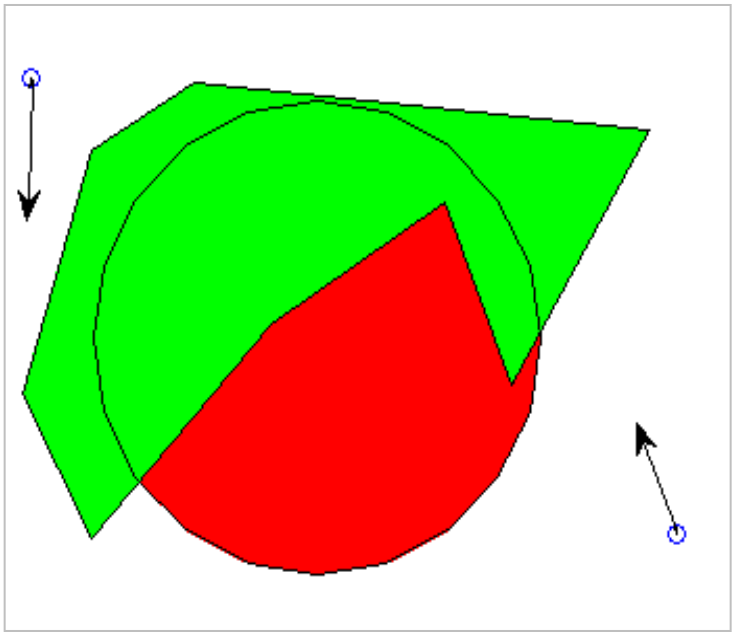


Figure 42 : Terrain 'Map01'

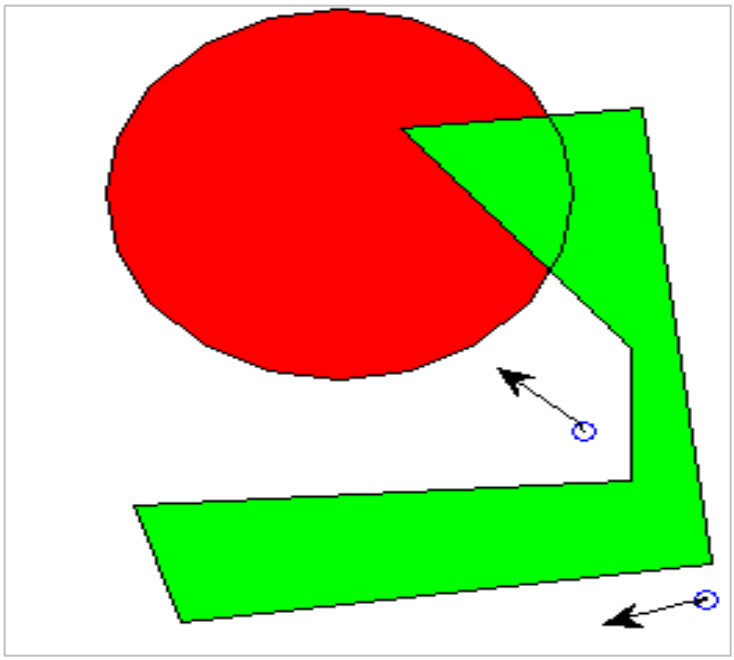


Figure 43 : Terrain 'Map02'

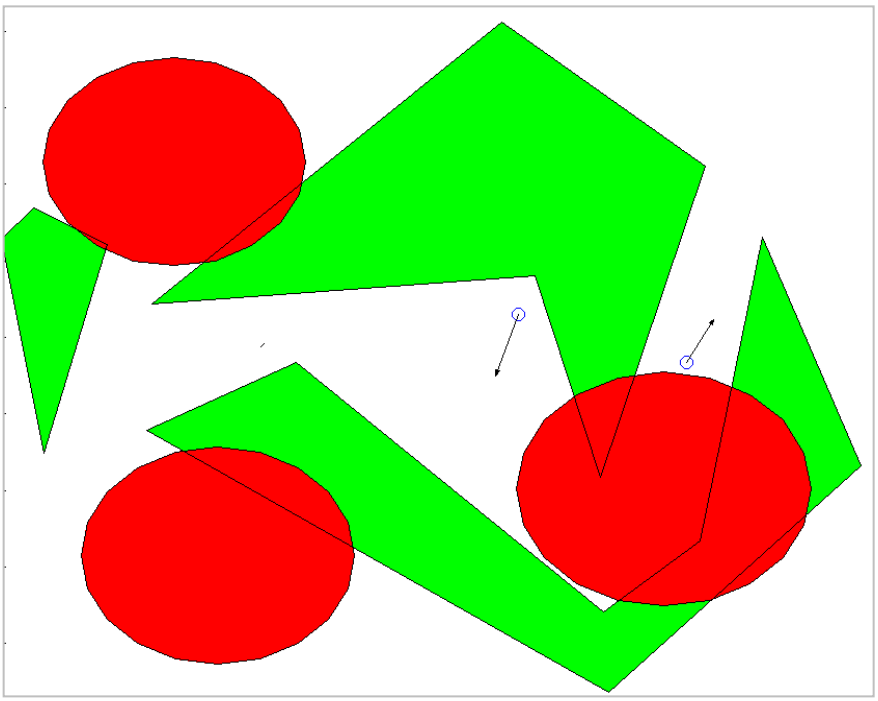


Figure 44 : Terrain 'Map03'

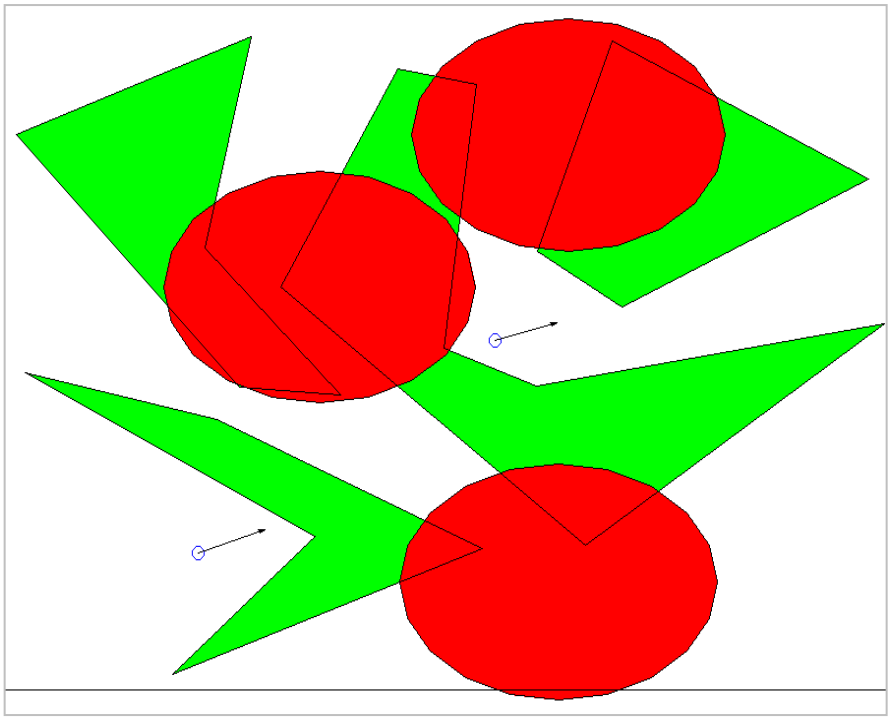


Figure 45 : Terrain 'Map04'

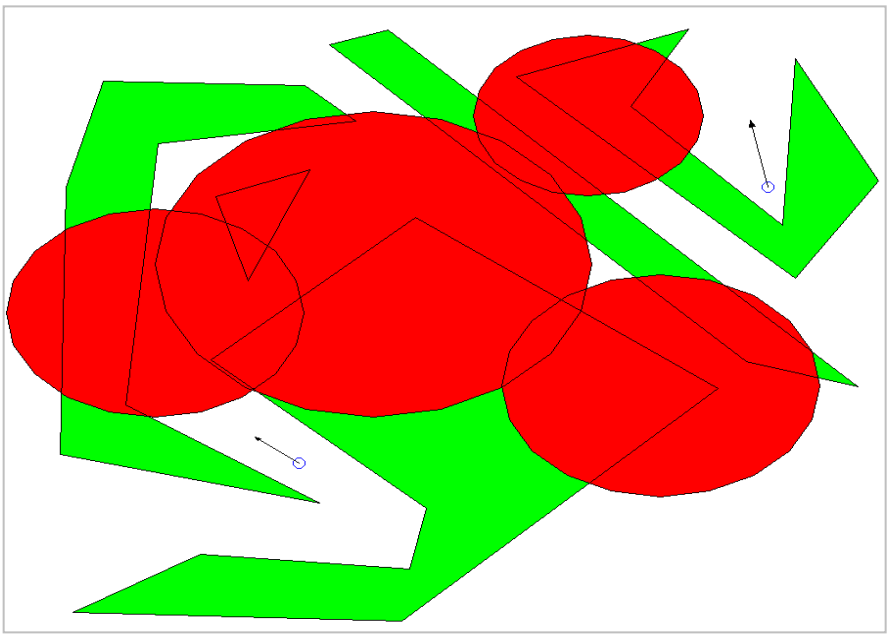


Figure 46 : Terrain 'Map05'

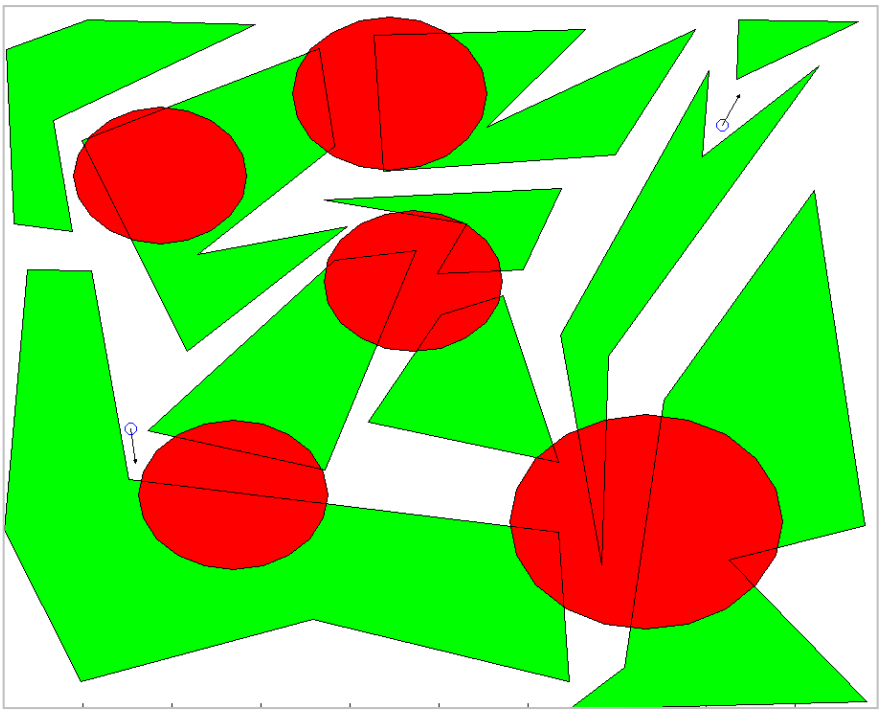


Figure 47 : Terrain 'Map06'

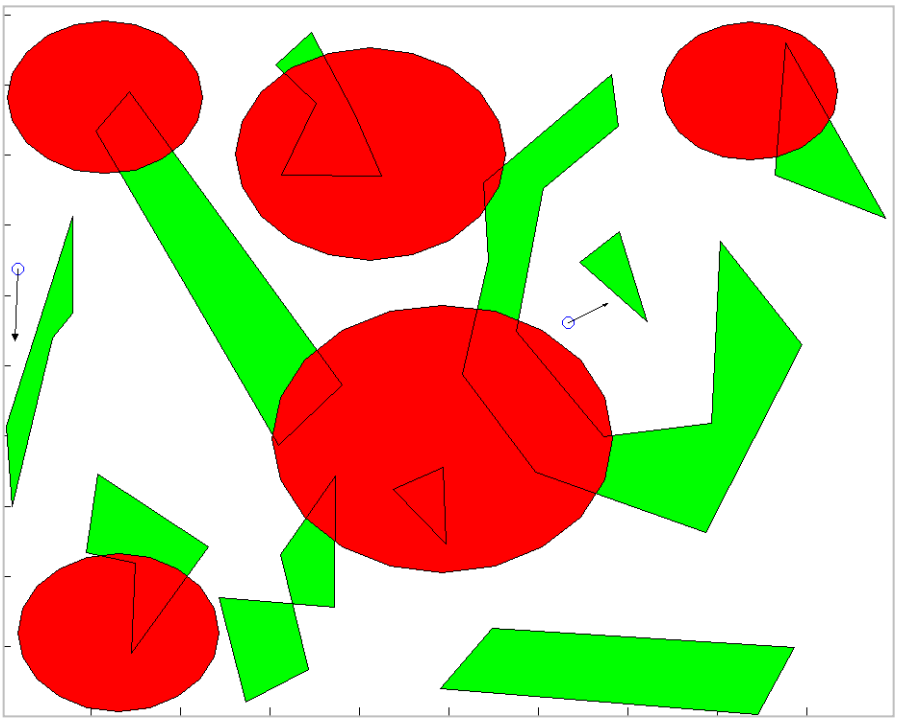


Figure 48 : Terrain 'Map07'

ANNEXE 3 – Les terrains utilisés pour la planification de la mission

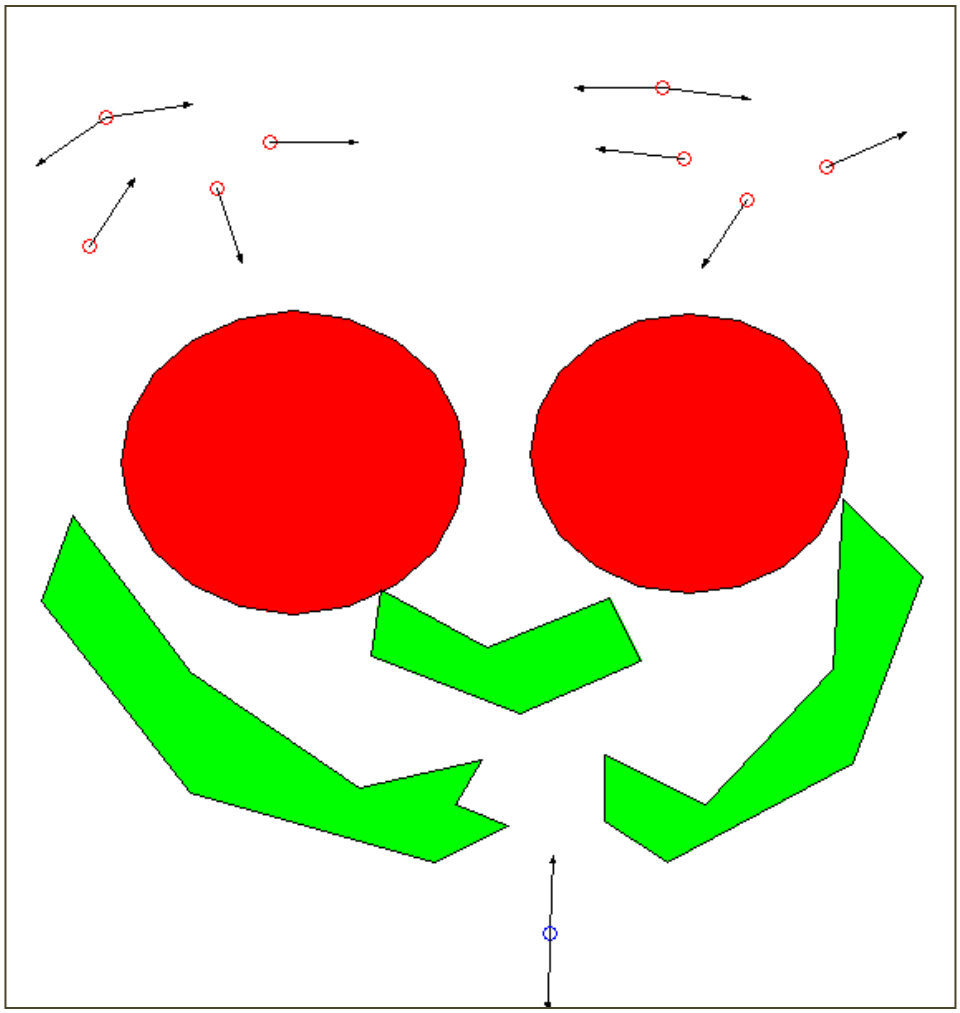


Figure 49 : Terrain 0 (3 obstacles, 2 zones de danger et 10 cibles)

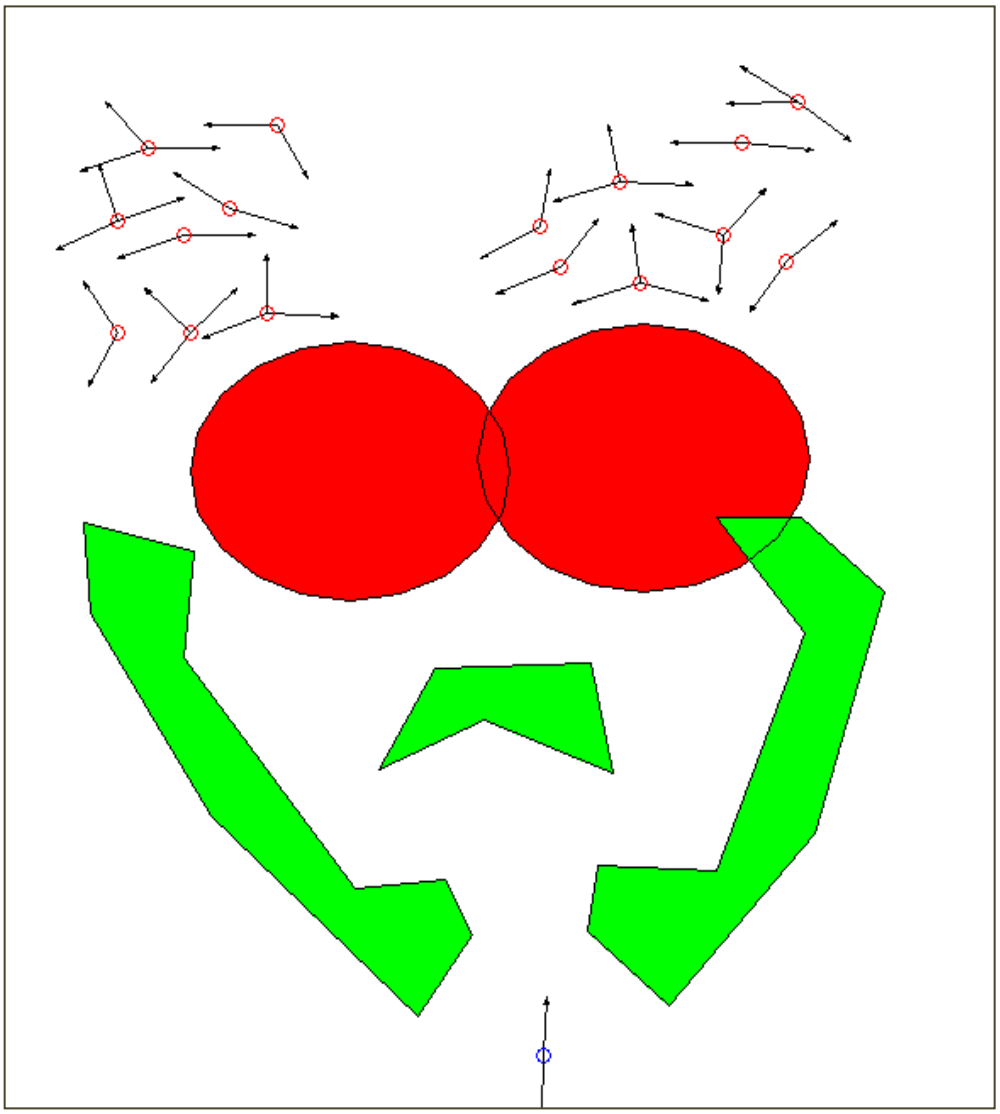


Figure 50 : Terrain 1 (3 obstacles, 2 zones de danger et 40 cibles)

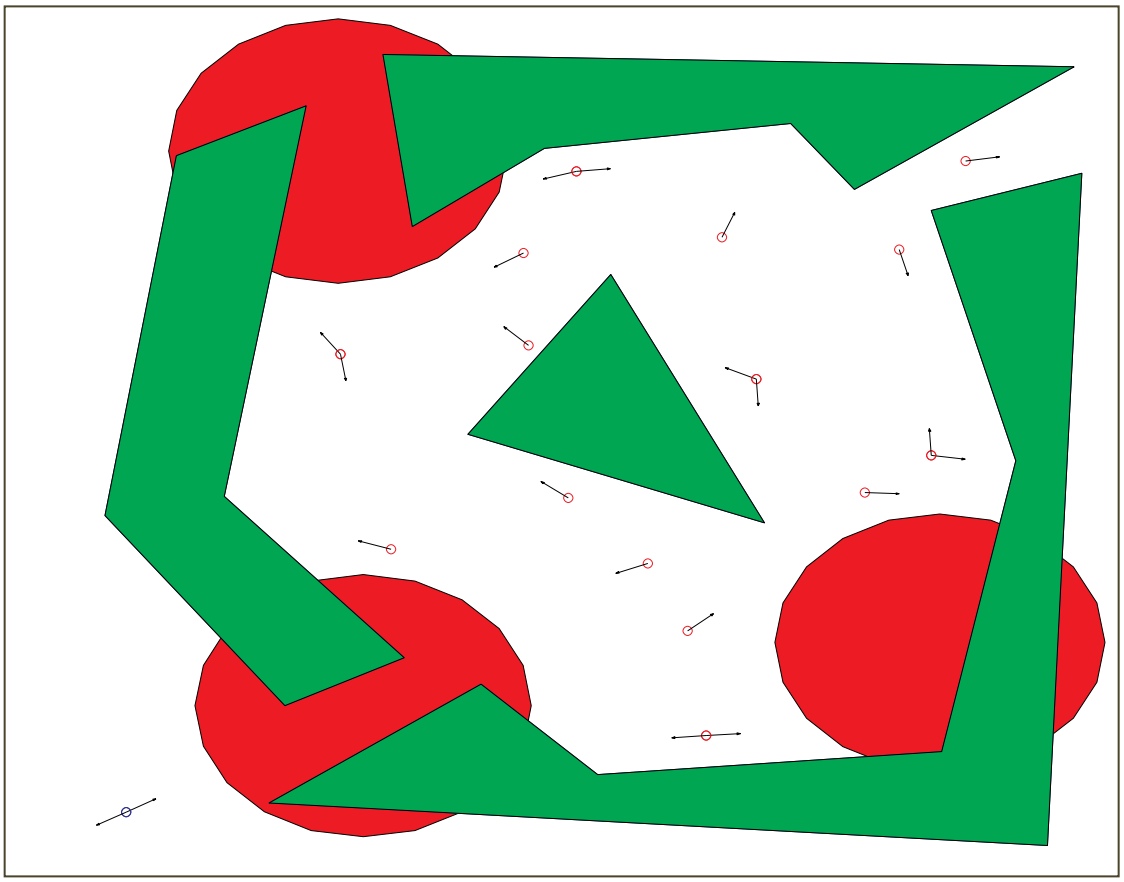
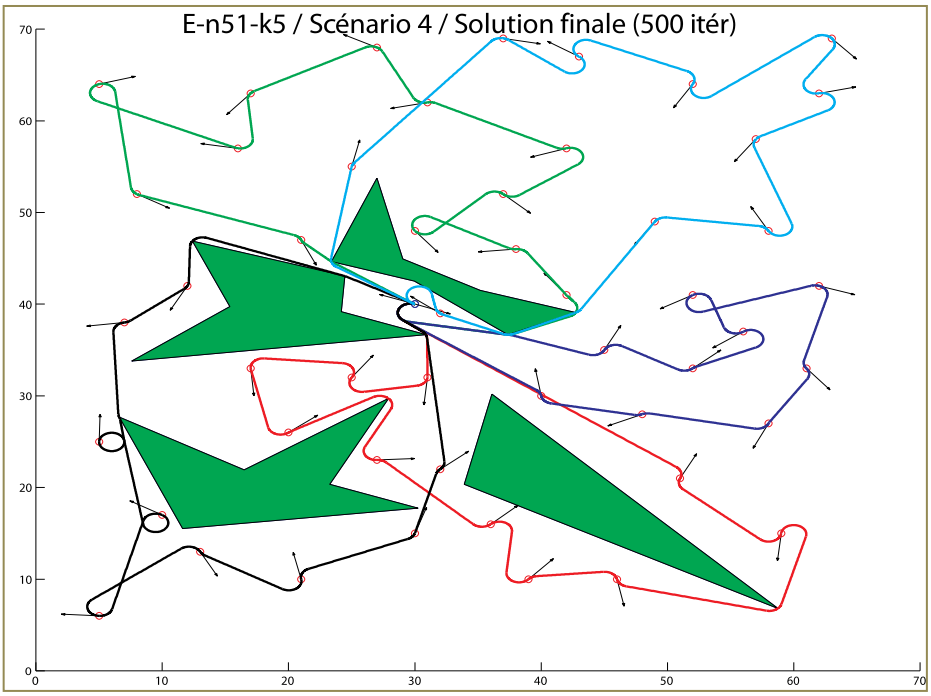


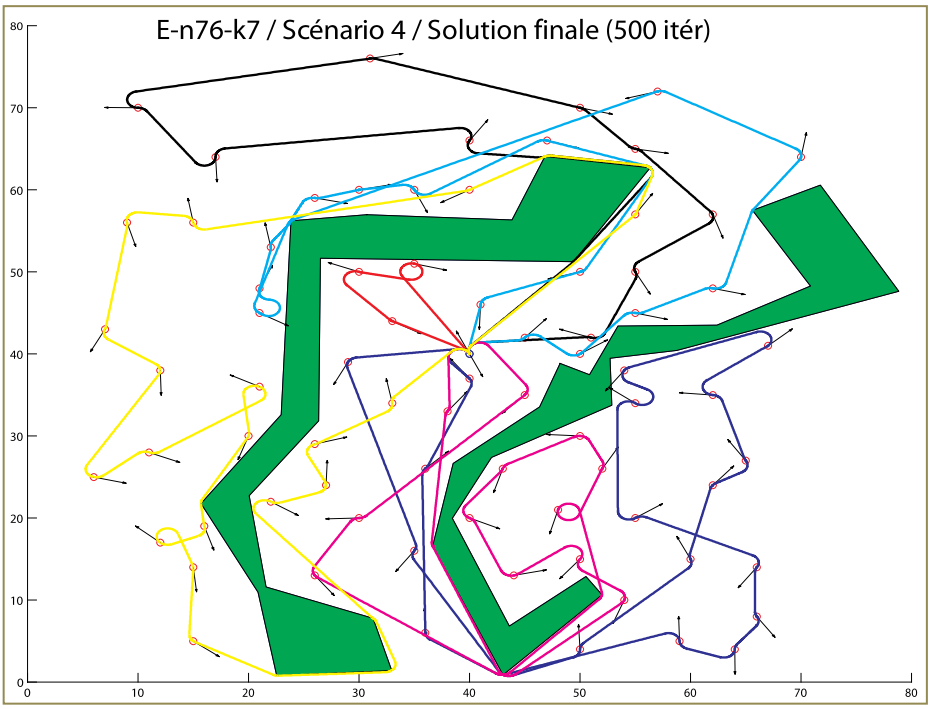
Figure 51 : Le terrain 2 (4 obstacles, 3 zones de danger et 20 cibles)

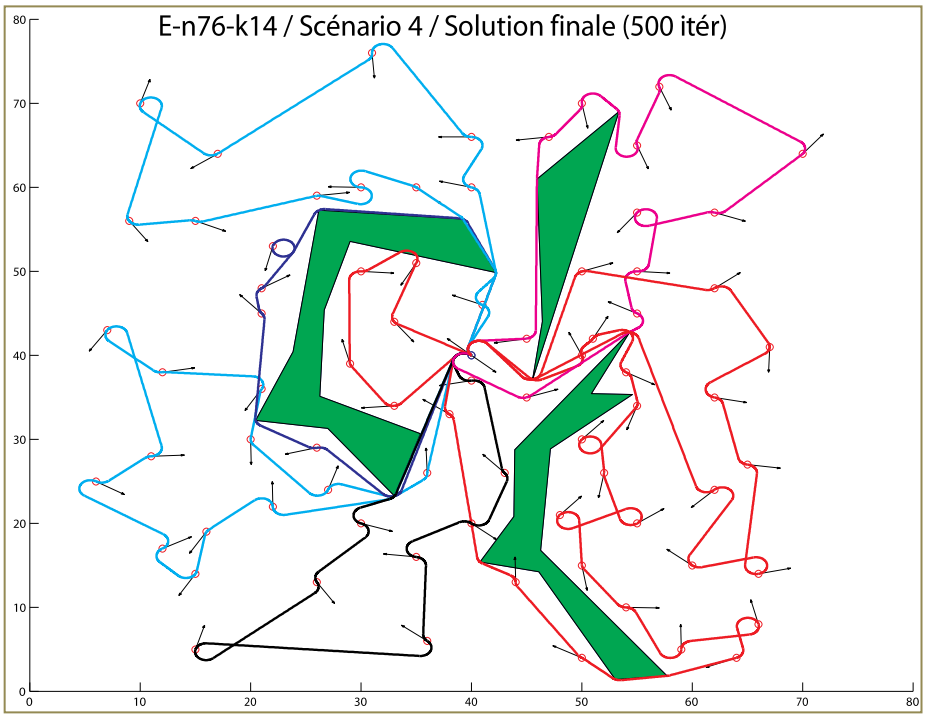
ANNEXE 4 – Extrait des itérations du Tabou pour le terrain 2 scénario 4

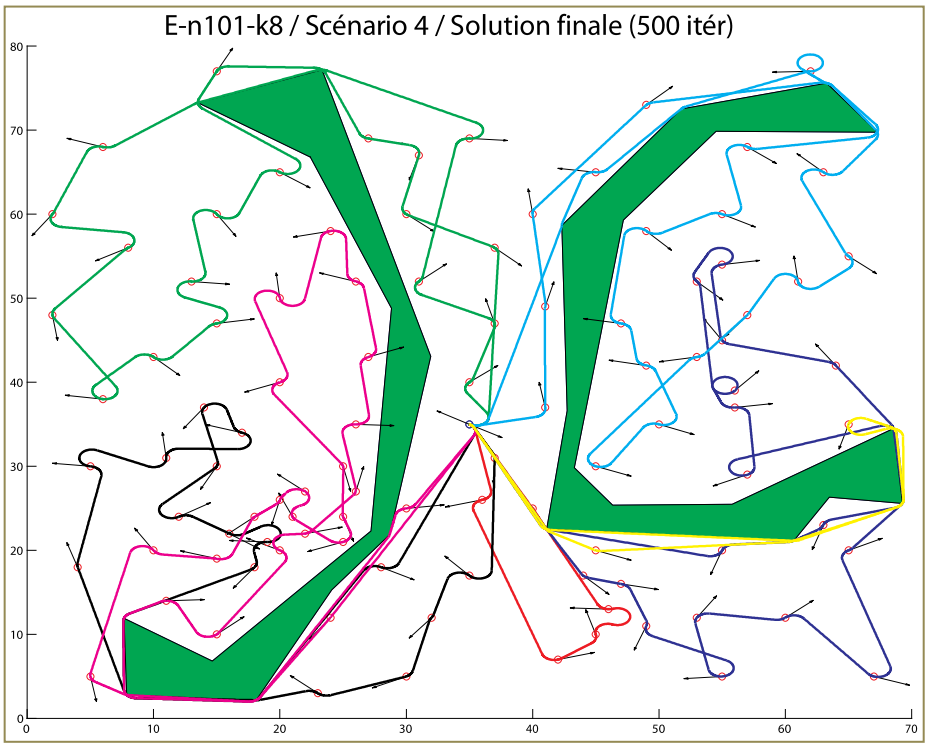
Iter	*TabooSize* (Inter, Intra)	*Movement*	*SolutionObj* (Initial,Good,Best,TheBest)	*Penalities* (Auto, risk)	*Best Sol* (Obj,Auton,Risk)
1	(0, 0)	[NaN, NaN, NaN]	[707.2, 707.2, 707.2, 707.2]	[1.000, 0.200]	[707.2, 248.2, 175.5]
2	(1, 0)	[7, 17, 18]	[707.2, 525.6, 525.6, 707.2]	[1.000, 0.200]	[707.2, 248.2, 175.5]
3	(1, 1)	[2, 17, 8]	[707.2, 512.0, 512.0, 707.2]	[1.000, 0.200]	[707.2, 248.2, 175.5]
4	(1, 2)	[1, 3, 4]	[707.2, 514.2, 512.0, 707.2]	[1.000, 0.200]	[707.2, 248.2, 175.5]
5	(1, 3)	[1, 12, 11]	[707.2, 516.5, 512.0, 707.2]	[1.000, 0.200]	[707.2, 248.2, 175.5]
6	(1, 4)	[1, 11, 19]	[707.2, 517.2, 512.0, 707.2]	[1.000, 0.200]	[707.2, 248.2, 175.5]
7	(1, 5)	[1, 13, 14]	[707.2, 519.9, 512.0, 707.2]	[2.000, 0.100]	[707.2, 248.2, 175.5]
8	(1, 6)	[2, 9, 10]	[707.2, 534.1, 518.9, 707.2]	[2.000, 0.100]	[707.2, 248.2, 175.5]
9	(1, 7)	[2, 9, 17]	[707.2, 550.2, 518.9, 707.2]	[2.000, 0.100]	[707.2, 248.2, 175.5]
10	(1, 8)	[2, 17, 16]	[707.2, 554.8, 518.9, 707.2]	[2.000, 0.100]	[707.2, 248.2, 175.5]
28	(16, 0)	[18, 7, 14]	[707.2, 697.1, 615.8, 697.1]	[16.000, 0.013]	[697.1, 243.2, 175.6]
29	(16, 1)	[2, 10, 9]	[707.2, 694.8, 615.8, 694.8]	[16.000, 0.013]	[694.8, 243.2, 175.6]
30	(16, 2)	[1, 4, 3]	[707.2, 692.7, 615.8, 692.7]	[16.000, 0.013]	[692.7, 241.1, 175.6]
95	(16, 0)	[10, 9, 2]	[707.2, 763.8, 615.8, 692.7]	[32.000, 0.013]	[692.7, 241.1, 175.6]
96	(17, 0)	[14, 10, 2]	[707.2, 676.5, 676.5, 676.5]	[32.000, 0.013]	[676.5, 248.8, 192.1]
97	(17, 1)	[1, 4, 3]	[707.2, 674.6, 674.6, 674.6]	[32.000, 0.006]	[674.6, 248.8, 192.1]
98	(17, 2)	[1, 19, 11]	[707.2, 673.6, 673.6, 673.6]	[32.000, 0.006]	[673.6, 248.8, 192.1]
99	(17, 3)	[1, 11, 12]	[707.2, 671.3, 671.3, 671.3]	[32.000, 0.006]	[671.3, 248.8, 192.1]
100	(17, 4)	[3, 9, 10]	[707.2, 670.3, 670.3, 670.3]	[32.000, 0.006]	[670.3, 248.8, 192.1]
150	(0, 0)	[21, 18, 6]	[707.2, 707.2, 707.2, 638.2]	[1.000, 0.200]	[638.2, 241.9, 191.4]
151	(1, 0)	[7, 17, 18]	[707.2, 525.6, 525.6, 638.2]	[1.000, 0.100]	[638.2, 241.9, 191.4]
152	(1, 1)	[2, 17, 8]	[707.2, 512.0, 512.0, 638.2]	[1.000, 0.100]	[638.2, 241.9, 191.4]
153	(1, 2)	[1, 3, 4]	[707.2, 514.2, 512.0, 638.2]	[1.000, 0.100]	[638.2, 241.9, 191.4]
154	(1, 3)	[1, 12, 11]	[707.2, 516.5, 512.0, 638.2]	[1.000, 0.100]	[638.2, 241.9, 191.4]
155	(1, 4)	[1, 11, 19]	[707.2, 517.2, 512.0, 638.2]	[2.000, 0.100]	[638.2, 241.9, 191.4]
495	(42, 0)	[22, 1, 10]	[707.2, 670.3, 640.4, 638.2]	[0.500, 0.000]	[638.2, 241.9, 191.4]
496	(43, 0)	[6, 17, 2]	[707.2, 660.4, 640.4, 638.2]	[0.500, 0.000]	[638.2, 241.9, 191.4]
497	(0, 0)	[22, 1, 16]	[707.2, 703.5, 703.5, 638.2]	[1.000, 0.200]	[638.2, 241.9, 191.4]
498	(1, 0)	[7, 8, 18]	[707.2, 512.0, 512.0, 638.2]	[1.000, 0.200]	[638.2, 241.9, 191.4]
499	(1, 1)	[1, 3, 4]	[707.2, 514.2, 512.0, 638.2]	[1.000, 0.100]	[638.2, 241.9, 191.4]
500	(1, 2)	[1, 12, 11]	[707.2, 516.5, 512.0, 638.2]	[1.000, 0.100]	[638.2, 241.9, 191.4]

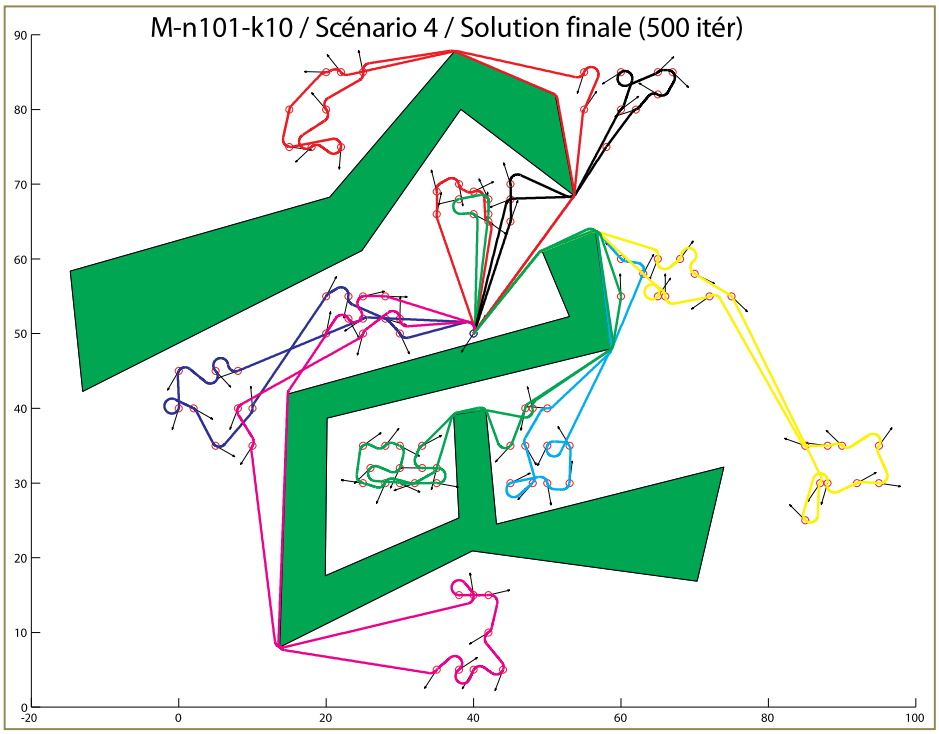
ANNEXE 5 – Exemple de résultats obtenus sur les instances du CVRP

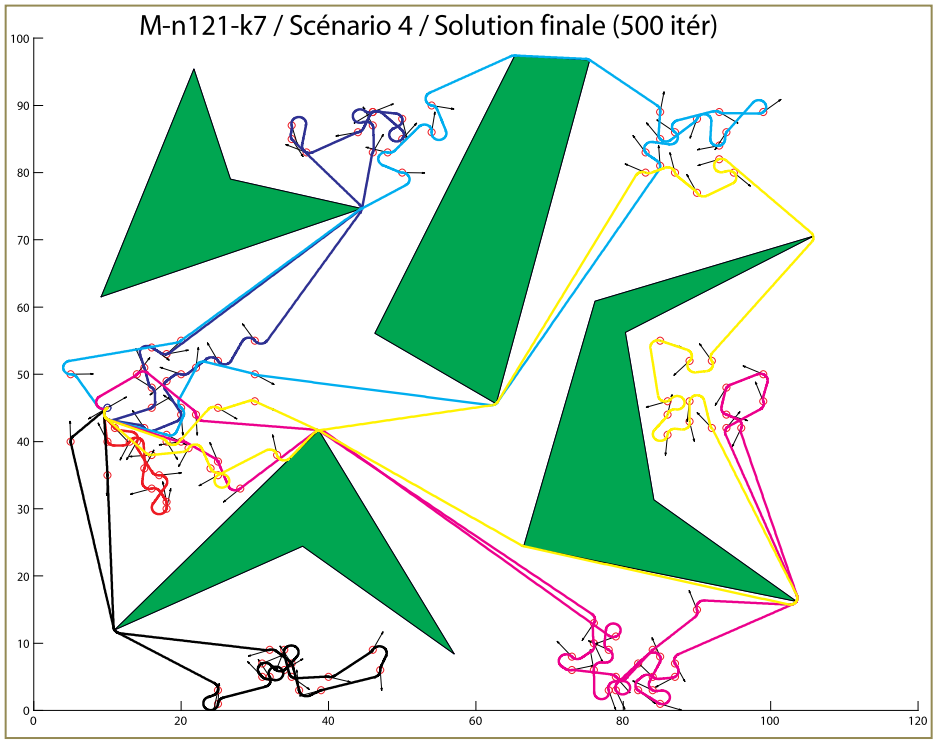


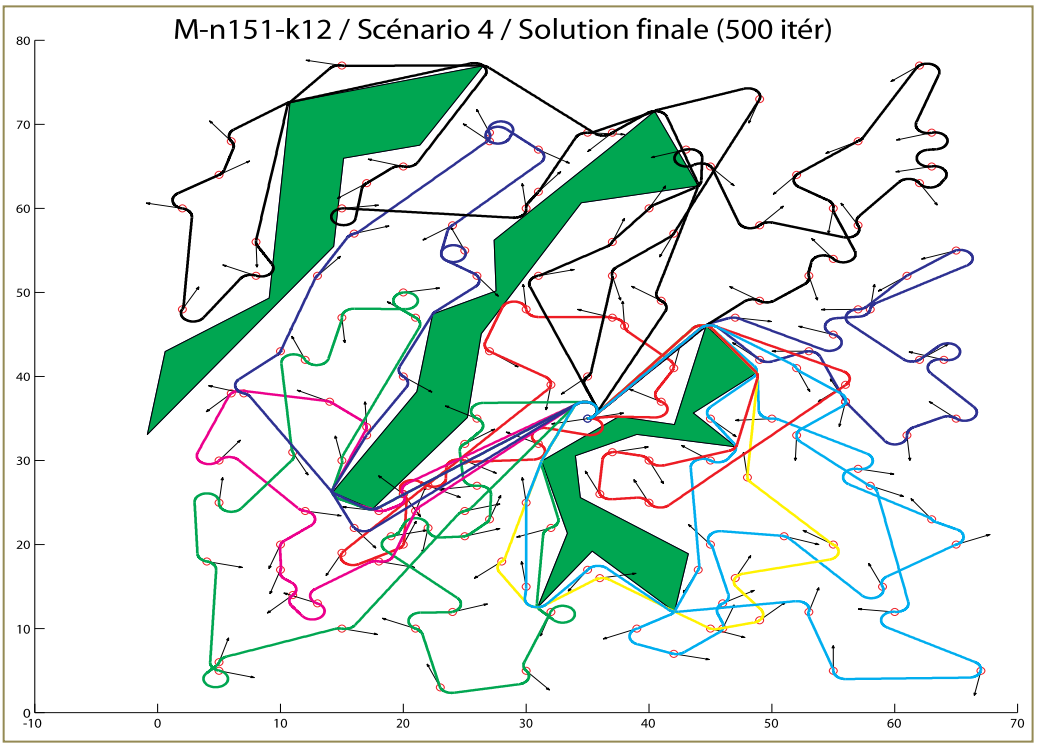


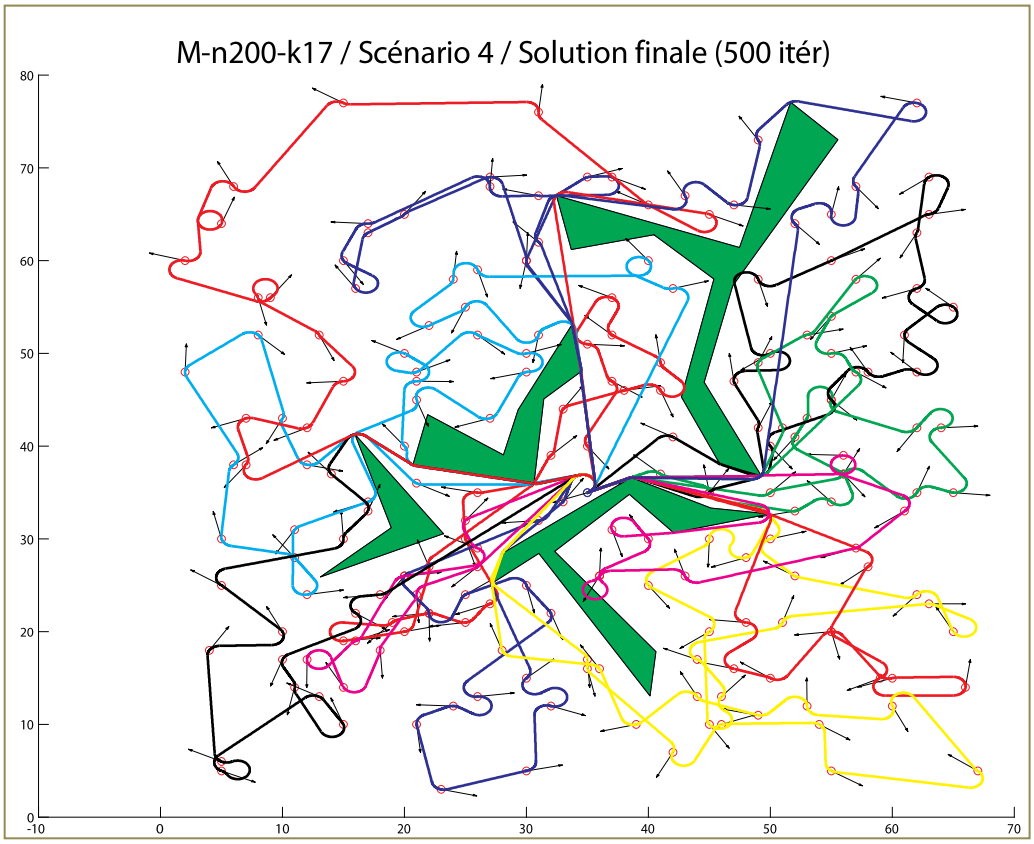

E-n76-k14 / Scénario 4 / Solution finale (500 itér)
0
10
20
30
40
50
60
70
80


E-n101-k8 / Scénario 4 / Solution finale (500 itér)
0
10
20
30
40
50
60
70
80


M-n101-k10 / Scénario 4 / Solution finale (500 itér)
90
80
70
60
50
40
30
20
10
0
-20
0
20
40
60
80
100


M-n121-k7 / Scénario 4 / Solution finale (500 itér)
100
90
80
70
60
50
40
30
20
10
0
0
20
40
60
80
100
120


M-n151-k12 / Scénario 4 / Solution finale (500 itér)
80
70
60
50
40
30
20
10
0
-10
0
10
20
30
40
50
60
70


M-n200-k17 / Scénario 4 / Solution finale (500 itér)
80
70
60
50
40
30
20
10
0
-10
0
10
20
30
40
50
60
70