



HAL
open science

Intégration des préférences d'un opérateur dans les décisions d'un drone autonome et élicitation incrémentale de ces préférences

Arwa Khannoussi

► To cite this version:

Arwa Khannoussi. Intégration des préférences d'un opérateur dans les décisions d'un drone autonome et élicitation incrémentale de ces préférences. Systèmes embarqués. Université de Bretagne occidentale - Brest, 2019. Français. ⟨NNT : 2019BRES0080⟩. ⟨tel-02536674⟩

HAL Id: tel-02536674

<https://theses.hal.science/tel-02536674v1>

Submitted on 8 Apr 2020

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

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



HAL Authorization

THESE DE DOCTORAT DE

L'UNIVERSITE
DE BRETAGNE OCCIDENTALE
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601
*Mathématiques et Sciences et Technologies
de l'Information et de la Communication*
Spécialité : *Informatique*

Par

Arwa KHANNOUSSI

Intégration des préférences d'un opérateur dans les décisions d'un drone autonome et élicitation incrémentale de ces préférences

Thèse présentée et soutenue à Brest, le 4 décembre 2019
Unité de recherche : Lab-STICC UMR CNRS 6285

Rapporteurs avant soutenance :

Vincent MOUSSEAU
Meltem ÖZTÜRK-ESCOFFIER

Professeur, CentraleSupélec
Maître de Conférences, HDR, Université Paris-Dauphine

Composition du jury :

Président : Frank SINGHOFF
Examineurs : Vincent MOUSSEAU
Meltem ÖZTÜRK-ESCOFFIER
Jacques PETIT-FRERE

Professeur, Université de Bretagne Occidentale
Professeur, CentraleSupélec
Maître de Conférences, HDR, Université Paris-Dauphine
Responsable activité Traitement de l'Information, Electronique
de Missiles, Thales

Dir. de thèse : Patrick MEYER
Encadr. : Catherine DEZAN

Professeur, IMT Atlantique
Maître de Conférences, Université de Bretagne Occidentale

Invité(s) :

Alexandru-Liviu OLTEANU
Jean-Philippe DIGUET (co-dir.)

Maître de Conférences, Université Bretagne Sud
Directeur de Recherche CNRS, Lab-STICC

*Cette thèse de doctorat est le fruit d'activités de recherche
menées dans Lateral, un laboratoire commun de recherche entre
Thales et le Lab-STICC (UMR CNRS 6285).*

Remerciements

Entamer une thèse ? L'idée paraît séduisante à quiconque privilégie l'épanouissement intellectuel. Mais les obstacles sont nombreux, et seules une solide motivation et l'aide de nombreuses personnes peuvent permettre d'en venir à bout... Et aujourd'hui en écrivant ces mots, je veux remercier ces personnes qui étaient là pour moi pour mener à bien ce beau projet.

En premier lieu, je tiens à exprimer ma gratitude et ma reconnaissance éternelles à toi *Patrick* mon "*Best Director Ever*". Je te remercie pour tes conseils, tes encouragements, ta disponibilité ainsi que pour ton investissement personnel qui témoigne de ton engagement à l'égard de l'avenir professionnel de tes doctorants. Le travail à tes côtés était très agréable et j'espère que ça va durer encore longtemps... Merci d'avoir cru en moi mais surtout d'avoir été à la fois mon chef et mon ami. Merci d'avoir subi mes peurs, mes angoisses, mes folies sans jamais me juger. Merci pour tous les délires et supers moments qu'on a pu partager.

Je tiens par ailleurs à remercier *Catherine Dezan*, *Jean-Philippe Diguet* pour leur encadrement.

Un grand merci à toi *Alexandru-Liviu Olteanu* pour ton implication dans la thèse. Merci *Alex* pour ta contribution, sans ton expertise ces travaux ne seraient pas aussi intéressants.

Un immense merci à *Sébastien Bigaret* et *Emmanuel Braux* pour leurs efforts sur la plateforme Open Stack qui m'ont permis de venir à bout de tous mes calculs et merci d'avoir toujours été présents (même pendant leurs vacances) pour régler les problèmes liés à mes instances de calcul.

Je tiens également à remercier *Meltem Öztürk-Escoffier* et *Vincent Mousseau* pour l'intérêt qu'ils ont porté à mes travaux de recherche, en me faisant l'honneur d'accepter la charge de rapporter cette thèse.

Je tiens aussi à remercier chaleureusement les "*précaires*" du département *LUSSE*, *Alex Kabil*, *Raphäel Charbey*, *Antoine Malléol*, *Vincent Dessarthe*, *Bertrand Charbonnier* et *Pierre Bégout* sans qui ces années de doctorat n'auraient pas été aussi agréables. Merci pour votre soutien et vos encouragements et surtout merci d'avoir été plus que des collègues, mais aussi des amis.

Je ne peux pas passer sans remercier tous mes amis. Un immense merci, en particulier à *Asma* et *Patrick* qui m'ont permis de ne jamais dévier de mon objectif final, à *Ines* pour les coups de fils pour m'encourager et me remonter le moral, à *Narymen*, *Amira*,...

Ces remerciements ne peuvent s'achever, sans une pensée à mes *chers parents*. Votre présence,

votre amour inconditionnel et vos encouragements sont pour moi les piliers fondateurs de ce que je suis et de ce que je fais. Que ce travail soit le témoignage de ma gratitude et de mon grand amour. Je remercie également mes sœurs pour leur soutien sans limites.

À tous ceux que j'aurais malencontreusement oubliés,

MERCI!

Table des matières

Remerciements	i
Table des matières	iii
Table des figures	vii
Liste des tableaux	xi
Introduction	xiii
Publications	xvii
I État de l’art	1
1 Drones autonomes	3
1.1 Introduction aux drones et aux missions	3
1.2 Décision pour les drones autonomes	8
1.2.1 Planification de la mission	8
1.2.2 Capteurs intelligents	10
1.2.3 Gestion des défauts	11
1.2.4 Décisions distribuées	12
1.3 Résumé et mise en perspective	12
2 Aide Multi-Critère à la Décision	15
2.1 Concepts généraux	15
2.2 Processus d’aide à la décision	17
2.3 Modélisation des préférences	19
2.3.1 Théorie de la valeur multi-attribut	20
2.3.2 Relations de surclassement	22
2.4 Apprentissage pour la décision multi-critère	25
2.4.1 Élicitation indirecte par lot des paramètres du modèle	26
2.4.2 Élicitation incrémentale	27
2.5 Résumé et mise en perspective	28

II	Intégration des préférences d'un opérateur dans les décisions d'un drone autonome	31
3	Moteur de décisions multi-niveaux	33
3.1	Processus de prise de décision d'un pilote humain	34
3.2	Moteur de décisions multi-niveaux	35
3.3	Intégration des préférences d'un pilote	38
3.3.1	Intégration des préférences dans le niveau 1	38
3.3.1.1	Modélisation du problème de décision	38
3.3.1.2	Instanciation dans notre moteur	40
3.3.1.3	Exemple illustratif	42
3.3.2	Intégration des préférences dans le niveau 2	44
3.3.2.1	Modélisation du problème de décision	44
3.3.2.2	Instanciation dans notre moteur	45
3.3.2.3	Exemple illustratif	47
3.3.3	Intégration des préférences dans le calcul de trajectoire	49
3.4	Résumé et mise en perspective	50
4	Simulateur de drone autonome	53
4.1	Motivations et choix	54
4.2	Modélisation du drone	55
4.3	Modélisation d'une mission	59
4.4	Interface utilisateur	63
4.5	Exemple d'exécution de mission	65
4.5.1	Exemple de mission	65
4.5.2	Exécution de la mission dans le simulateur	69
4.6	Résumé et mise en perspective	77
III	Élicitation par lot et incrémentale des paramètres de modèles SRMP	79
5	Élicitation par lot des paramètres de modèles SRMP	81
5.1	État de l'art et étude d'expressivité	82
5.1.1	Programme linéaire mixte en nombres entiers	82
5.1.2	Expressivité des modèles SRMP	86
5.1.3	Problème de satisfaisabilité booléenne (SAT)	88
5.1.4	Matheuristique	88
5.2	Proposition d'un algorithme génétique	90
5.2.1	Généralités	90
5.2.2	Proposition des variantes d'un algorithme génétique	92
5.2.3	Étude expérimentale	96
5.2.3.1	Configurations étudiées	96
5.2.3.2	Expériences	97
5.2.3.3	Résultats	97
5.2.4	Algorithme génétique d'apprentissage	105
5.2.5	Comportement de l'algorithme génétique face à de nouvelles données	105
5.3	Résumé et mise en perspective	106

6	Élicitation incrémentale des paramètres de modèles SRMP	109
6.1	Élicitation incrémentale des préférences	110
6.1.1	Apprentissage actif et élicitation de préférences incrémentale	110
6.1.2	Processus d'élicitation incrémentale	111
6.1.3	Résolution des incohérences	114
6.2	Heuristiques de sélection	119
6.2.1	Heuristiques utilisant l'ensemble des alternatives	120
6.2.1.1	\mathcal{H}_{rnd} : heuristique aléatoire	120
6.2.1.2	\mathcal{H}_{sim} : heuristique d'alternatives similaires	120
6.2.1.3	\mathcal{H}_{dis} : heuristique d'alternatives dissimilaires	120
6.2.2	Heuristiques utilisant l'ensemble des solutions	120
6.2.2.1	\mathcal{H}_{cp} : heuristique d'alternatives proches d'un profil	121
6.2.2.2	\mathcal{H}_{fp} : heuristique d'alternatives loin d'un profil	121
6.2.2.3	\mathcal{H}_{mp} : heuristique d'alternatives maximisant le nombre de profils nécessaires pour les discriminer	121
6.3	Configuration du MIP	122
6.3.1	$\mathcal{MIP}_{\text{first}}$: premier modèle SRMP réalisable	123
6.3.2	$\mathcal{MIP}_{\text{close}}$: modèle SRMP le plus proche du modèle précédent	123
6.3.3	$\mathcal{MIP}_{\text{center}}$: un modèle SRMP placé au centre de l'espace de recherche	124
6.4	Validation empirique	124
6.4.1	Plan d'expériences	125
6.4.2	Résultats expérimentaux et discussion	125
6.4.3	En pratique	133
6.5	Élicitation incrémentale avec l'algorithme génétique	134
6.6	Résumé et mise en perspective	138
	Conclusion et perspectives	141
	Résumé	142
	Discussion et limites	143
	Perspectives	144
	Bibliographie	147
	Annexes	159
A	Résultats détaillés de l'algorithme génétique	159
A.1	Résultats des test de la configuration 1	159
A.2	Résultats des test de la configuration 2	162
A.3	Résultats des test de la configuration 3	166
A.4	Résultats des test de la configuration 4	169
B	Les programmes linéaires mixte en nombres entiers	173
C	Temps d'exécution de l'algorithme génétique	177

Table des figures

3.1	Moteur de décisions multi-niveaux	36
3.2	Intégration des préférences de l'opérateur dans le moteur de décisions	39
3.3	Présentation des événements dans le modèle MR-Sort	43
3.4	Exemple de modèle SRMP	48
4.1	Thales Watchkeeper WK450 [Thales, 2010]	55
4.2	Les forces qui agissent sur le drone en croisière.	56
4.3	Les forces qui agissent sur le drone en montée.	57
4.4	Les forces qui agissent sur le drone en inclinaison latérale.	58
4.5	La vitesse réelle du drone.	59
4.6	Moteur de décisions multi-niveaux	64
4.7	Carte de risque pour le drone.	67
4.8	Carte de risque pour l'environnement.	68
4.9	Modèle SRMP de l'opérateur #1	69
4.10	Modèle SRMP de l'opérateur #2	70
4.11	Simulation de la mission avec les préférences de l'opérateur #1	71
4.12	Simulation de la mission avec les préférences de l'opérateur #2	72
4.13	Représentation des 5 actions dans le modèle SRMP de l'opérateur #1	73
4.14	Représentation des 5 actions dans le modèle SRMP de l'opérateur #2	74
4.15	Représentation des 5 actions dans le modèle SRMP de l'opérateur #1	76
4.16	Représentation des 5 actions dans le modèle SRMP de l'opérateur #2	77
5.1	Les étapes d'un algorithme génétique	91
5.2	Évaluation (ϕ) moyenne pour les problèmes <i>3P5C</i> avec la configuration #1	98
5.3	Évaluation (ϕ) moyenne pour les problèmes <i>2P7C</i> avec la configuration #4	99
5.4	Évaluation (ϕ) moyenne pour les problèmes <i>2P5C</i> avec la configuration #3 et $P_{mut} = 0.3$	100
5.5	Évaluation (ϕ) moyenne pour les problèmes <i>3P3C</i> avec la configuration #1 et $P_{mut} = 0.7$	101
5.6	Évaluation (ϕ) moyenne avec la configuration #4 et $n_{pop} = 100$	102
5.7	Évaluation (ϕ) moyenne avec la configuration #1 et $n_{pop} = 100$	103
5.8	Évaluation (ϕ) moyenne pour les différentes configurations avec $n_{pop} = 100$ et $P_{mut} = 0.7$	104
5.9	Tau de <i>Kendall</i> moyen pour les différents problèmes	106
6.1	Processus d'élicitation incrémentale général	112
6.2	Processus d'élicitation incrémentale en cas d'incohérence de préférences	113
6.3	Processus d'élicitation incrémentale	119

6.4	Processus d'élicitation incrémentale	123
6.5	Moyenne et écart type du tau de <i>Kendall</i> de l'heuristique \mathcal{H}_{rnd} pour les problèmes $2P3C$	126
6.6	Moyenne du tau <i>Kendall</i> des heuristiques pour $\mathcal{MIP}_{\text{first}}$ pour les problèmes $2P3C$	127
6.7	Moyenne du tau <i>Kendall</i> pour les problèmes $2P3C$ et $2P5C$	128
6.8	Moyenne du tau de <i>Kendall</i> de \mathcal{H}_{mp} pour les problèmes $2P3C$ et $2P5C$	129
6.9	Temps d'exécution pour les problèmes $2P3C$	130
6.10	Temps d'exécution pour les problèmes $2P5C$	131
6.11	Moyenne du tau de <i>Kendall</i> de \mathcal{H}_{mp} et \mathcal{H}_{rnd} pour les problèmes $2P7C$	132
6.12	Tau de <i>Kendall</i> moyen et temps d'exécution moyen cumulé de $\mathcal{H}_{\text{mp}} / \mathcal{MIP}_{\text{first}}$ pour les problèmes $2P3C$	133
6.13	Tau de <i>Kendall</i> moyen et temps d'exécution moyen cumulé de $\mathcal{H}_{\text{mp}} / \mathcal{MIP}_{\text{first}}$ pour les problèmes $2P5C$	134
6.14	Moyenne et écart type du tau de <i>Kendall</i> avec l'algorithme génétique pour les problèmes $2P3C$	135
6.15	Moyenne du tau de <i>Kendall</i> avec l'algorithme génétique pour les différents problèmes	136
6.16	Le temps d'exécution moyen et l'écart type avec l'algorithme génétique (AG) pour les problèmes $2P3C$ et $3P7C$	137
6.17	Moyenne du tau de <i>Kendall</i> avec l'algorithme génétique (AG) et le MIP pour les problèmes $2P3C$ et $2P5C$	138
A.1	Évaluation (ϕ) moyenne pour les problèmes $2P3C$	159
A.2	Évaluation (ϕ) moyenne pour les problèmes $2P5C$	160
A.3	Évaluation (ϕ) moyenne pour les problèmes $2P7C$	160
A.4	Évaluation (ϕ) moyenne pour les problèmes $3P3C$	161
A.5	Évaluation (ϕ) moyenne pour les problèmes $3P5C$	161
A.6	Évaluation (ϕ) moyenne pour les problèmes $3P7C$	162
A.7	Évaluation (ϕ) moyenne pour les problèmes $2P3C$	163
A.8	Évaluation (ϕ) moyenne pour les problèmes $2P5C$	163
A.9	Évaluation (ϕ) moyenne pour les problèmes $2P7C$	164
A.10	Évaluation (ϕ) moyenne pour les problèmes $3P3C$	164
A.11	Évaluation (ϕ) moyenne pour les problèmes $3P5C$	165
A.12	Évaluation (ϕ) moyenne pour les problèmes $3P7C$	165
A.13	Évaluation (ϕ) moyenne pour les problèmes $2P3C$	166
A.14	Évaluation (ϕ) moyenne pour les problèmes $2P5C$	167
A.15	Évaluation (ϕ) moyenne pour les problèmes $2P7C$	167
A.16	Évaluation (ϕ) moyenne pour les problèmes $3P3C$	168
A.17	Évaluation (ϕ) moyenne pour les problèmes $3P5C$	168
A.18	Évaluation (ϕ) moyenne pour les problèmes $3P7C$	169
A.19	Évaluation (ϕ) moyenne pour les problèmes $2P3C$	170
A.20	Évaluation (ϕ) moyenne pour les problèmes $2P5C$	170
A.21	Évaluation (ϕ) moyenne pour les problèmes $2P7C$	171
A.22	Évaluation (ϕ) moyenne pour les problèmes $3P3C$	171
A.23	Évaluation (ϕ) moyenne pour les problèmes $3P5C$	172
A.24	Évaluation (ϕ) moyenne pour les problèmes $3P7C$	172
C.1	Le temps d'exécution moyen et l'écart type pour les problèmes $2P5C$	177

C.2	Le temps d'exécution moyen et l'écart type pour les problèmes $2P\ 7C$	178
C.3	Le temps d'exécution moyen et l'écart type pour les problèmes $3P\ 3C$	178
C.4	Le temps d'exécution moyen et l'écart type pour les problèmes $3P\ 5C$	179

Liste des tableaux

1.1	Les niveaux d'autonomie d'un drone.	6
3.1	Les paramètres du modèle MR-Sort	42
3.2	Évaluations des alternatives et les paramètres du modèle SRMP	47
3.3	Encodage des évaluations des alternatives	48
4.1	Spécifications du WK450	55
4.2	Les coordonnées géographiques des waypoints	65
4.3	Paramètres de préférences du premier opérateur.	68
4.4	Paramètres de préférences du deuxième opérateur.	69
4.5	Évaluation des actions au waypoint 1.	70
4.6	Évaluation des actions au waypoint 5.	75
5.1	MIP pour inférer un modèle SRMP	85
5.2	La moyenne et l'écart-type (en pourcentage) de la répartition des paires d'alternatives par rapport au nombre de profils nécessaires à leur construction	87
5.3	La moyenne et l'écart-type (en pourcentage) de la répartition des paires d'alternatives dominées par rapport au nombre de profils nécessaires à leur construction.	88
5.4	Différentes configurations de l'algorithme génétique.	96
6.1	MP_2 pour déterminer les contraintes incompatibles dans l'inférence d'un modèle SRMP	116
6.2	Liste des préférences/indifférences du décideur	117
6.3	Évaluations des alternatives	118
6.4	Fonction de pénalité de \mathcal{H}_{mp}	122
6.5	Temps d'exécution global approximatif en jours pour un test exhaustif des problèmes $2P\ 5C$	132
B.1	MIP pour inférer le modèle SRMP le plus proche	174
B.2	MIP pour inférer les modèles SRMP d'extrémité	175
B.3	MIP pour inférer un modèle SRMP central	176

Introduction

Un drone est un véhicule aérien (aéronef) qui ne transporte pas de pilote humain. Dans un aéronef classique, le rôle de ce dernier est d'observer, de réagir et de prendre des décisions à bord afin de mener à bien une mission. Dans un drone, son absence implique donc une réduction des capacités sensorielles et de prise de décision de l'aéronef. En conséquence, le taux de réussite de missions effectuées par des drones est en général inférieurs à celui des aéronefs avec pilote. En effet, des facteurs comme les changements météorologiques, les erreurs liées aux informations sur l'environnement de vol, les menaces inconnues et les conséquences des défaillances des sous-systèmes des véhicules sont des difficultés que les pilotes humains ont l'habitude de gérer, mais qui dépassent régulièrement les compétences de la plupart des premiers drones développés.

Dans un drone, le pilote est remplacé par un système qui lui apporte un certain niveau d'autonomie et lui permet de reproduire certaines capacités sensorielles et décisionnelles d'un pilote humain. Le niveau d'autonomie d'un drone tient en général compte de trois facteurs : l'indépendance à l'égard de l'engagement humain, la complexité de la mission et celle de l'environnement. Classiquement, trois niveaux d'autonomie sont cités [Huang, 2004] : (1) pilotage à distance, c'est à dire que le drone est télécommandé par un opérateur au sol, (2) semi-autonome, c'est-à-dire que le drone reçoit des ordres de haut niveau et il est surveillé par un opérateur au sol, (3) entièrement autonome, c'est-à-dire que le drone est capable d'accomplir une mission sans l'intervention d'un opérateur humain et de prendre des décisions de façon totalement autonome lorsqu'il en a besoin.

Dans cette thèse nous nous intéressons aux drones entièrement autonomes. Au cours des dernières décennies, l'endurance de vol de ces drones a fortement augmenté et ils peuvent aujourd'hui réaliser des missions de plus en plus longues et complexes. Cela sous-entend que le plan de la mission, préparé en amont, doit potentiellement pouvoir être changé ou mis à jour à cause de l'apparition d'événements extérieurs (modification de l'environnement, ...) ou internes (liés aux sous-systèmes du drone) pendant le vol. Dans ce cas le drone autonome doit être capable de prendre des décisions de façon complètement autonome, sans l'intervention d'un opérateur humain. Dans la pratique cela sous-entend également que l'opérateur, qui au mieux ne fait que surveiller le drone, doit avoir une confiance élevée dans les décisions du drone. L'objectif principal de cette thèse est donc de proposer un moteur de décisions à embarquer dans le drone autonome qui garantisse un niveau de confiance élevé de l'opérateur dans la capacité du drone à prendre les « bonnes » décisions.

Dans le contexte de ce travail nous avons identifié trois facteurs sur lesquels nous pouvons influencer pour augmenter la confiance de l'opérateur dans les décisions du drone. Tout d'abord, il est important que la structure du moteur de décisions ainsi que le fonctionnement des algorithmes embarqués puissent être facilement expliqués à l'opérateur, afin d'éviter tout effet de « boîte noire ». Ensuite, un opérateur aura plus de confiance dans un drone si son comportement

pendant la mission est celui auquel il s'attend. Pour terminer, il est important que les décisions prises par le drone puissent être interprétées sans trop de difficultés. Ce dernier aspect sous-entend notamment qu'il doit être possible de retracer l'origine d'une décision dans le moteur, et de comprendre pourquoi elle a été prise (en vue peut-être de la rectifier pour une prochaine mission, si elle ne convient pas à l'opérateur).

C'est pour cette raison que nous faisons l'hypothèse ici que la prise de décision d'un drone autonome devrait être basée sur la logique adoptée par un pilote humain lors d'une mission. Le processus de prise de décision d'un pilote humain est classiquement décrit par le cycle OODA (« Observer, Orienter, Décider et Agir ») décrit par le pilote John Boyd [Fadok, 1995]. De manière synthétique, ce cycle représente le processus de décision d'un humain qui doit prendre des décisions dans un environnement aéronautique avec des conditions extérieures changeantes. *Observer*, consiste à prendre note de certaines caractéristiques de l'environnement, *orienter* consiste à pointer son avion vers l'adversaire, afin d'être en bonne position pour entrer l'étape. *Décider* consiste à choisir que faire ensuite, et, *agir* consiste à mettre en œuvre ce qui a été décidé.

Ceci signifie donc que pendant le vol le pilote se retrouve régulièrement face à des problèmes de décision complexes où il doit réfléchir aux alternatives possibles, à leur impact sur la mission, et à leur exécution. En général, le pilote décompose ce problème en un ensemble de sous-problèmes, en adoptant l'approche « diviser pour mieux régner » [Armstrong *et al.*, 1975]. Il commence par choisir une action de haut niveau parmi un ensemble d'actions possibles (par exemple, atterrir, sauter un point de passage, retourner à la base, ...). Le choix de l'une ou l'autre de ces actions se base sur leur évaluation selon plusieurs critères (risque, coût, objectif de la mission) qui déterminent leur impact sur la mission. Ce choix dépend aussi de la façon opérationnelle dont l'action peut être exécutée (choix de la trajectoire, ...), ce qui constitue en soi une sous-décision. Il est intéressant de noter que dans une même situation, le résultat de ce processus de prise de décision peut varier d'un pilote à l'autre, selon son niveau d'expertise, ses expériences et ses propres jugements de la situation.

Ces observations constituent le point de départ de ces travaux, dans lesquels nous proposons un moteur de décisions multi-niveaux à embarquer dans un drone autonome qui imite le processus de décision d'un pilote humain pendant le vol. Il est composé de deux niveaux de décisions principaux. Le premier niveau permet de surveiller l'état du drone et de son environnement pour détecter les événements qui peuvent perturber la réalisation de la mission et déclencher la prise de décision du second niveau. Ce dernier une fois déclenché permet de choisir une action de haut niveau (atterrir, continuer, ...) la mieux adaptée à la situation courante parmi un ensemble d'actions possibles. Un des principaux avantages d'un tel moteur multi-niveaux est l'explicabilité accrue de son fonctionnement, et un traçage simplifié des décisions, deux éléments qui contribuent à renforcer la confiance que l'opérateur devrait avoir dans les décisions du drone autonome.

Une autre particularité du moteur de décisions que nous proposons est qu'il permet d'intégrer les préférences de l'opérateur au sol, en utilisant des techniques issues du domaine de l'Aide Multi-Critères à la Décision (AMCD) [Roy, 1985], dans ses différents niveaux. Cela doit permettre au drone de prendre des décisions selon de multiples critères, tout en prenant en compte les priorités et les préférences de l'opérateur.

Dans ce travail nous menons ensuite une discussion argumentée concernant les différentes classes de modèles d'AMCD qui peuvent être intégrées dans les différents niveaux du moteur, afin de garantir un haut niveau d'explicabilité du modèle et d'interprétabilité de la décision.

Nous argumentons ensuite, le choix d'utiliser un modèle de tri multi-critère particulier (MR-Sort) [Bouyssou et Marchant, 2007a] pour le niveau 1 et un modèle de classement multi-critère spécifique (SRMP) [Rolland, 2013a] pour le niveau 2 dans notre contexte applicatif.

Afin d'illustrer l'intérêt de notre moteur de décisions multi-niveaux et montrer que les décisions prises par le drone dépendent bien du profil de l'opérateur (c'est à dire de ses préférences) nous développons un simulateur de drone autonome. Ce simulateur modélise un « Thales Watchkeeper WK450 » [Thales, 2010] qui doit effectuer des missions définies par une série de points de passages. Nous détaillons les fonctionnalités du simulateur, et présentons l'implémentation pratique d'une mission de surveillance au dessus de la ville de Brest en France. Cet exemple nous permet de souligner que l'exécution de la mission varie selon les préférences de deux opérateurs différents, et nous mettons en avant l'interprétabilité des décisions prises par le drone.

Les modèles d'AMCD partagent une caractéristique commune : ils nécessitent une phase en amont de la mission, lors de laquelle les préférences de l'opérateur sont élicitées, avant d'être intégrées dans le drone. Nous nous intéresserons donc dans cette thèse à l'élicitation des préférences des modèles SRMP embarqués dans le second niveau du moteur. Pour réduire l'effort cognitif de l'opérateur pendant la phase d'élicitation des préférences, nous proposons donc un processus d'élicitation incrémental. Il permet de gérer dans certains cas les incohérences dans les préférences exprimées par le décideur.

Dans notre étude de ce processus incrémental, nous cherchons d'abord à trouver une heuristique qui permet de choisir les questions à soumettre à l'opérateur pour obtenir un modèle représentant au mieux ses préférences. Ces questions sont sous la forme de paires d'alternatives (donc dans notre cas des décisions de haut niveau auxquelles le drone est soumis) au sujet desquelles il ou elle doit exprimer une indifférence ou une préférence stricte. Le choix de ces paires est crucial, afin de converger le plus rapidement vers un modèle satisfaisant, pour minimiser l'effort cognitif de l'opérateur.

Pour déterminer une bonne heuristique de choix des paires à soumettre à l'opérateur, nous utilisons la programmation linéaire mixte en nombres entiers (MIP) pour éliciter les paramètres préférentiels du modèle SRMP à chaque itération. Nous remarquons cependant que les temps de calcul à chaque itération sont potentiellement très élevés, et que par conséquent un tel processus est incompatible avec beaucoup d'applications réelles. Cela nous motive ensuite à proposer une méthode approchée d'identification des préférences, plus précisément un algorithme génétique, pour résoudre ce problème de temps d'exécution. Cependant, comme le montre la validation empirique que nous entreprenons, l'utilisation d'une méthode approchée nécessite un plus grand nombre de paires d'apprentissage pour atteindre un modèle SRMP aussi satisfaisant que le MIP.

Afin d'exposer nos travaux, nous avons choisi de structurer ce manuscrit en trois parties. La partie **I**, qui englobe 2 chapitres, est consacrée à l'état de l'art. Le but du chapitre **1** est de se familiariser avec le contexte des drones autonomes et d'explorer les différents aspects de la prise de décision dans cette situation. Le chapitre **2** présente les concepts généraux d'AMCD, les approches pour modéliser les préférences d'un décideur humain, ainsi que la façon de les éliciter.

La partie **II** est dédiée au moteur de décisions du drone autonome que nous proposons ainsi qu'à l'intégration des préférences d'un opérateur dans ces décisions. Elle débute, avec le chapitre **3** qui présente notre proposition de moteur de décisions multi-niveaux à embarquer dans le drone. Ce moteur imite le processus de décision d'un pilote, tout en intégrant ses préférences.

Dans le chapitre 4 nous détaillons le simulateur de drone autonome que nous avons développé. Il modélise le « Thales Watchkeeper WK450 » par un point qui est soumis à des contraintes et conditions physiques semblables au vrai drone. Dans ce simulateur le drone intègre le moteur de décisions multi-niveaux et il permet de souligner l'intérêt de notre approche.

La partie III est consacrée à l'élicitation incrémentale des modèles SRMP. Elle commence par le chapitre 5 qui propose différentes méthodes exactes et approchées pour l'élicitation par lot des modèles SRMP. Finalement, le chapitre 6 présente notre étude qui mène à la proposition d'un processus d'élicitation incrémentale des modèles SRMP, dont le but de réduire l'effort cognitif de l'opérateur du drone dans la phase d'élicitation des préférences en amont de la mission.

Nous terminons ce manuscrit par un chapitre consacré à la synthèse de nos résultats, ainsi qu'à une analyse des perspectives qui découlent de nos travaux.

Publications scientifiques

La plupart des chapitres de ce travail sont basés sur des publications présentées à des conférences ou des ateliers détaillés ci-dessous. Elles représentent nos contributions actuelles dans les domaines de l'aide multi-critère à la décision ainsi que l'autonomie des drones.

Articles dans des actes de conférences avec comité de lecture

- [Khannoussi *et al.*, 2019] A. Khannoussi, A.-L. Olteanu, C. Labreuche, P. Narayan, C. Dezan, J.-P. Diguët, J. Petit-Frère, P. Meyer, Integrating Operators' Preferences into Decisions of Unmanned Aerial Vehicles : Multi-layer Decision Engine and Incremental Preference Elicitation, ADT'2019 : Algorithmic Decision Theory, Oct 2019, Durham, NC, USA.
- [Khannoussi *et al.*, 2018b] A. Khannoussi, A.-L. Olteanu, C. Dezan, J.-P. Diguët, C. Labreuche, J. Petit-Frère, P. Meyer, Incremental Learning of Simple Ranking Method Using Reference Profiles Models, DA2PL'2018 : from Multiple Criteria Decision Aid to Preference Learning, Nov 2018, Poznan, Poland.

Communications à des séminaires et des conférences sans actes

- [Khannoussi *et al.*, 2018c] A. Khannoussi, A.-L. Olteanu, C. Dezan, J.-P. Diguët, P. Meyer, Incremental preference elicitation for SRMP models : Application for autonomous drones, 88th Meeting of the EURO Working Group Multiple Criteria Decision Aiding, 27-28 September 2018, Lisbon, Portugal, 2018
- [Khannoussi *et al.*, 2018a] A. Khannoussi, P. Meyer, C. Dezan, J.-P. Diguët, A.-L. Olteanu, Traceable decisions for autonomous unmanned aerial vehicles, ROADEF 2018, Lorient, France, February 2018
- [Khannoussi *et al.*, 2017] A. Khannoussi, P. Meyer, C. Dezan, J.-P. Diguët : Integrating human preferences in automated decisions of unmanned aerial vehicles, 14th Decision Deck Workshop, 20 September 2017, Paris, France, 2017

Article dans une revue internationale avec comité de lecture[\[khannoussi et al. en révision\]](#)

A. Khannoussi, A.-L. Olteanu, C. Dezan, J.-P. Diguët, C. Labreuche, J. Petit-Frère, P. Meyer, Incremental Learning of Simple Ranking Method Using Reference Profiles Models, *4OR* (**en révision**)

Première partie

État de l'art

Chapitre 1

Drones autonomes

Sommaire

1.1 Introduction aux drones et aux missions	3
1.2 Décision pour les drones autonomes	8
1.2.1 Planification de la mission	8
1.2.2 Capteurs intelligents	10
1.2.3 Gestion des défauts	11
1.2.4 Décisions distribuées	12
1.3 Résumé et mise en perspective	12

Les travaux présentés dans ce manuscrit se concentrent principalement sur la prise de décision dans les drones autonomes durant une mission. Pour cette raison, dans ce chapitre, nous commençons par présenter les drones nous donnons les éléments qui les caractérisent, et nous introduisons les notions importantes pour notre travail. Ensuite, une définition de la mission est donnée dans le contexte des drones ainsi que des exemples de différents types de missions. Ce qui mène à présenter les travaux réalisés pour la prise de décision des drones autonomes durant les missions. Ces travaux sont regroupés en 4 catégories : la planification de la mission, les capteurs intelligents, la gestion des défauts et les décisions distribuées. Pour terminer, nous expliquons notre positionnement par rapport à l'état de l'art.

1.1 Introduction aux drones et aux missions

Un drone (en anglais Unmanned Aerial Vehicle, UAV) est défini comme un appareil motorisé capable de se déplacer dans l'air (un aéronef) qui ne transporte pas de pilote humain, qui peut voler de façon autonome ou être télécommandé à distance et peut transporter une charge utile [SD, 2005].

Au cours des dernières décennies, des efforts considérables ont été consacrés à l'augmentation de l'endurance de vol et de la charge utile des drones, ce qui s'est traduit par diverses configurations de drones de différentes tailles, niveaux d'endurance et de capacités. Cela permet de proposer de nouveaux drones qui répondent au mieux à ces nouvelles exigences. Il existe diverses façons de les classer (par exemple, configuration aérodynamique, taille, . . .). [Dalamagkidis, 2015] les classes en fonction de leur technique de vol dans l'une des cinq catégories suivantes :

- **Les drones à aile fixe** font référence aux aéronefs sans pilote avec des ailes qui nécessitent une piste pour décoller et atterrir, ou un lancement en catapulte. Ils ont généralement une grande endurance et peuvent voler à des vitesses élevées.
- **Les drones à voilure tournante ou de giravion**, aussi appelés drones à décollage et atterrissage verticaux, ont l'avantage de pouvoir voler en stationnaire et d'être très maniables. Ces capacités sont utiles pour de nombreuses missions robotiques, notamment dans les applications civiles. Un drone de giravion peut avoir différentes configurations, avec un rotor principal et en arrière (hélicoptère classique), un rotor coaxial, rotor en tandem, multirotors, etc.
- **Les dirigeables**, qui sont plus légers que l'air et ont une grande endurance, volent à basse vitesse et sont généralement de grande taille.
- **Les drones à ailes battantes**, qui ont de petites ailes souples inspirées par les oiseaux et les insectes volants.
- **Les drones à configurations hybrides ou convertibles**, qui peuvent décoller verticalement et incliner leurs rotors ou leur corps et voler comme des avions à aile fixe.

Le fait de retirer le pilote humain d'un aéronef enlève beaucoup de sa capacité sensorielle et de prise de décision. L'absence d'un pilote pour observer, réagir et prendre des décisions à bord du drone a entraîné des taux de perte plus élevés, au niveau de la réussite de la mission et au niveau des aéronefs, que ceux des véhicules pilotés. Des facteurs comme les changements météorologiques, les erreurs dans les informations sur l'environnement de vol, les menaces inconnues et les conséquences des défaillances des sous-systèmes des véhicules sont des difficultés que les pilotes humains ont l'habitude de gérer, mais qui dépassent la capacité de la plupart des premiers drones développés.

Pour qu'un drone puisse faire preuve d'un haut niveau d'autonomie sans l'intervention d'un humain, un degré plus élevé d'autonomie à bord est nécessaire pour reproduire certaines des capacités sensorielles et décisionnelles d'un pilote humain. De plus, en cas de perte de communication, l'inclusion de l'autonomie à bord peut permettre au drone de poursuivre ses opérations en toute sécurité ou de retourner à un endroit prédéfini.

[Spitzer, 2001] présente un drone comme un système complexe composé de six sous-modules principaux qui travaillent en coordination pour combler l'absence à bord du pilote :

- **La cellule du drone** est une plate-forme simple, légère, aérodynamiquement efficace et stable avec un espace limité pour l'avionique, et évidemment pas de place pour un pilote ;
- **L'ordinateur de bord**, le cœur du drone, est un système informatique conçu pour recueillir des informations aérodynamiques à partir d'un ensemble de capteurs (accéléromètres, gyroscopes, magnétomètres, capteurs de pression, GPS, etc.), afin de diriger automatiquement le vol d'un avion selon son plan de vol via plusieurs gouvernes présentes sur la cellule. ;
- **La charge utile** est un ensemble de capteurs composé de caméras, de capteurs infrarouges, de capteurs thermiques, etc. pour recueillir des informations qui peuvent être partiellement traitées à bord ou transmises à une station de base pour d'autres applications ;
- **Le contrôleur de mission/charge utile** est un ordinateur à bord du drone qui doit contrôler le système et le fonctionnement des capteurs inclus dans la charge utile. Cette

opération doit être effectuée conformément à l'élaboration du plan de vol ainsi que la mission réelle assignée au drone ;

- **La station de base** est un système informatique au sol conçu pour suivre l'évolution de la mission et éventuellement contrôler le drone et sa charge utile ;
- **L'infrastructure de communication** est un mélange de mécanismes de communication (modems radio, satellites, liaisons hertziennes, etc.) qui doivent garantir la liaison continue entre le drone et la station de base.

Nous observons donc que le « système drone » s'étend au-delà de l'aéronef, et inclut une base au sol qui permet de surveiller et de contrôler l'aéronef, ainsi qu'un système de communication. C'est pour ça que récemment, les organisations internationales comme l'Organisation de l'aviation civile internationale (OACI), EUROCONTROL, l'Agence européenne de la sécurité aérienne (AESA), la Federal Aviation Administration (FAA) ainsi que le Département de la défense des États-Unis (DoD) ont adopté le terme officiel de système aérien sans pilote (en anglais Unmanned Aerial System, UAS). Cet acronyme reflète ce changement de paradigme en mettant en avant les trois éléments principaux qui caractérisent le système :

- Le terme « sans pilote » fait référence à l'absence d'un pilote à bord ;
- Le terme « aérien » signifie qu'il s'agit d'un aéronef ;
- Le terme « système » est introduit pour signifier que l'UAS n'est pas seulement un véhicule, mais un système (distribué) composé d'une station de contrôle au sol, de liaisons de communication et des systèmes de lancement et de récupération en plus du véhicule lui-même.

Dans ce manuscrit de thèse, nous utilisons le terme de « *pilote* » pour désigner une personne à bord du drone et le terme d'« *opérateur* » pour une personne au sol simplement chargée de surveiller ou télécommander le drone à distance.

Le pilote est remplacé par un système qui permet d'assurer un niveau d'autonomie au drone. Le niveau d'autonomie est basé sur trois aspects caractérisant l'autonomie : le niveau de l'indépendance à l'égard de l'engagement humain, de la complexité de la mission et de l'environnement. En 2002, les niveaux de contrôle autonome ont été proposés pour mesurer l'autonomie. Plus précisément, dix de ces niveaux ont été proposés dans [Clough, 2002] et sont fondés sur des exigences comme la compréhension de la situation, l'analyse, la coordination, la prise de décision et la capacité opérationnelle. Dans le tableau 1.1 nous présentons ces niveaux d'autonomie de 0 à 10, 0 étant le niveau le plus bas c'est-à-dire que l'opérateur décide de tout et 10 le niveau le plus haut où le drone est complètement autonome et l'opérateur n'intervient dans aucune décision.

Certaines des distinctions entre ces niveaux ne s'appliquent pas à tous drones. [Huang, 2004] a proposé une classification plus simple qui ne tient compte que du niveau de participation humaine et qui est compatible avec les quatre modes opérationnels. Elle est composée des trois classes suivantes :

- **Pilotage à distance** : un opérateur certifié contrôle le drone à distance à travers une liaison de données à vue directe ou avec les retours de ses capteurs ;
- **Semi-autonome** : le drone reçoit des ordres de haut niveau (les points de passage (way-points en anglais), l'objet à suivre, . . .), et sa performance est surveillée par un opérateur

Niveau d'autonomie	Description
0	Véhicule télécommandé.
1	Exécuter une mission pré-planifiée.
2	Une mission modifiable.
3	Réponse limitée aux défauts/événements en temps réel.
4	Réponse pour anticiper les erreurs/événements.
5	Véhicule adaptatif en cas de panne/événement.
6	Coordination multi-véhicules en temps réel.
7	Coopération multi-véhicules en temps réel.
8	Optimisation de la performance des missions multi-véhicules.
9	Optimisation de la performance tactique multi-véhicules.
10	Totalement autonome.

Tableau 1.1 – Les niveaux d'autonomie d'un drone.

formé à cet effet. Dans ce cas, le vol est effectué par le drone lui-même, mais toutes les décisions sont déléguées à un humain ;

- **Entièrement autonome** : le drone est chargé des tâches générales et il est capable de déterminer comment les accomplir, même face à des événements imprévus. Il peut également surveiller son état de santé et prendre des mesures correctives après l'apparition de défaillances. Le drone peut être surveillé à distance par un opérateur au sol.

Les drones sont devenus un pilier des opérations militaires et civiles actuelles. Ils sont préférés aux avions avec pilotes parce qu'ils réduisent les risques et augmentent la probabilité de la réussite de la mission tout en évitant de perdre des humains si la mission échoue. Une autre raison est la meilleure capacité de vigilance des drones par rapport aux humains lors des longues missions critiques, car il n'y a pas de risque de fatigue. C'est pour ces raisons que les drones sont souvent utilisés dans des missions dites « 3D » (pour Dangerous, Dirty or Dull en anglais) [Barrie, 1997, Marshall, 2004] c'est-à-dire des missions longues, sales et dangereuses. Les longues missions sont les opérations qui nécessitent plus que de 30 ou 40 heures de mission avec des tâches simples en termes de charge de travail. Il est préférable de les effectuer avec des drones parce que de telles tâches peuvent être simplement automatisées et ne nécessitent souvent qu'une surveillance humaine plutôt qu'un contrôle direct et continu. Le côté sale d'une mission est relié au fait qu'elle est exécutée dans un environnement nuisible à l'équipage comme par exemple les missions d'échantillonnage ou d'observation aériennes pour des catastrophes liées aux armes chimiques. Une mission est dite dangereuse lorsque le niveau de risque est trop élevé pour l'équipage humain.

Si nous prêtons attention aux applications civiles, un large éventail de scénarios apparaît. Par exemple, les opérations de recherche et de sauvetage [Waharte et Trigoni, 2010] où les drones peuvent être bénéfiques pour étudier l'environnement et recueillir des preuves sur la position d'une personne disparue plus rapidement puisque le critère temps est critique dans ce cas.

Il y a également eu des progrès dans l'utilisation des drones autonomes dans le domaine de

l'agriculture [Ju et Son, 2018]. En effet, ils jouent un rôle important pour relever certains des défis de l'agriculture moderne. En particulier, il peuvent assurer la durabilité de la productivité agricole qui est devenue difficile à maintenir en raison du changement climatique et ainsi répondre à la demande croissante de produits agricoles.

Une autre application en pleine croissance pour les drones civils est la reconstitution en 3 dimensions [Wang et Li, 2008] des bâtiments, zones et des objets. Les images prises par le drone à partir de différents angles sont utilisées pour extraire des informations de géométrie et de texture pour reconstruire des modèles 3D. D'autres applications environnementales [Jensen, 2007] se focalisent sur la collecte des données sur la terre, les océans et l'atmosphère. Ces données sont utilisées pour des prévisions météorologiques, la cartographie des différentes zones (urbaines, agricoles, des forêts, ...), la détection des phénomènes géologiques, ...

Ces dernières années, la complexité de ces missions a considérablement augmenté ce qui a entraîné une transition des missions à drone unique vers les missions à multiples drones. Les missions actuelles à multiples drones se classent en général en deux scénarios opérationnels possibles [Roldán et al., 2015]. Dans le premier scénario, plusieurs drones exécutent ensemble une seule tâche afin d'améliorer l'efficacité et le temps d'exécution. Par exemple dans le cas de transport des charges lourdes [Maza et al., 2009] où il faut transporter une grande quantité de médicaments vers des zones isolées. Dans le second scénario, plusieurs drones peuvent effectuer différentes tâches qui sont liées entre elles sans forte dépendance, mais avec un objectif de mission global commun, partagé. Un bon exemple de ce scénario est la mission de surveillance et de reconnaissance [Pham et al., 2008], qui consiste à surveiller une zone et à détecter des cibles potentielles. Chaque tâche peut être effectuée par des capteurs appropriés installés sur différents drones, ce qui permet d'améliorer l'efficacité de la mission.

Toutes ces missions et quelques soit leur domaine d'application ou leur type sont définies par une structure [Roldán et al., 2015]. Cette structure est composée de deux aspects. Le premier, de haut niveau, est lié aux perspectives de la mission. Dans cet aspect, une mission est alors vue comme un ensemble de tâches qui utilisent des ressources pour accomplir des objectifs. Le deuxième aspect est lié au fait que la mission soit une opération robotique. Pour cela une mission peut être vue comme un ensemble de tâches où chaque tâche est composée d'un ensemble de sous-tâches/actions. Cela mène à décrire une mission de la façon suivante :

- une **mission** est un ensemble d'objectifs qui nécessitent des ressources.
exemple : contrôler un incendie dans une ville;
- un **objectif** est le résultat à atteindre pour la réussite de la mission.
exemple : surveiller l'incendie dans la zone X, éteindre l'incendie dans la zone Y;
- une **ressource** est un élément (drone, capteur, personnes, objet, lieu, ...) disponible pour atteindre un objectif ou accomplir une tâche.
exemple : drone, caméra, policier et aéroport;
- une **tâche** est un ensemble d'actions réalisées pour atteindre un objectif.
exemple : surveillance incendie, extinction incendie;
- une **action** est une manœuvre de drone ou l'utilisation d'un capteur.
exemple : aller à un point de passage (latitude, longitude, altitude, temps) (waypoint en anglais) et prendre une photo;

- une **observation** est une information qui peut être utilisée pour déterminer l'état de la tâche.

exemple : l'altitude du drone est supérieure à 1000 mètres et le niveau du carburant du drone est bas.

Ces composants vont nous permettre dans la suite du travail de définir nos propres missions, selon nos besoins.

Il existe plusieurs techniques qui permettent de modéliser une mission pour définir sa spécification, mais aussi son déroulement. Nous pouvons citer quelques modèles utilisés pour les missions des drones. Le modèle de Markov est présenté dans [Zhu, 1991] pour donner un aspect d'adaptabilité durant la mission. [Barbier et Chantry, 2004] utilisent les réseaux de Petri dans des applications de contrôle et de suivi de mission. Les machines d'état sont utilisées dans [Belta et al., 2007] pour la spécification des missions. Le choix d'une technique de modélisation appropriée est fondamental non seulement pour la spécification de la mission, mais aussi pour le contrôle et la surveillance de celle-ci. Cette technique doit produire des modèles appropriés pour diverses tâches (par exemple, surveillance et reconnaissance) qui comprennent des états réguliers (par exemple, décollage, vol jusqu'à un point et atterrissage) et exceptionnels (par exemple, urgence, pannes et menaces).

1.2 Décision pour les drones autonomes

Pour s'assurer qu'un drone a toujours le même niveau de sécurité qu'un aéronef piloté par un pilote humain, les capacités sensorielles et de prise de décision doivent être automatisées. Pour cela les drones doivent posséder un niveau d'« intelligence » plus élevé en s'inspirant des qualités d'un pilote humain. Le drone doit acquérir la capacité de surveiller ses systèmes internes ainsi que le monde extérieur, et de détecter tout changement ayant une incidence sur la sécurité et les résultats de la mission. Avec ces informations le drone doit prendre les décisions nécessaires pour préserver la sécurité et atteindre les objectifs de la mission. Nous avons identifié que cette capacité pouvait être mise en œuvre grâce à l'utilisation d'une architecture de commande dite intelligente où la décision est présente dans les différents éléments de l'autonomie à bord du drone : la planification de la mission, les capteurs intelligents, la gestion des défauts et les décisions distribuées. Nous allons présenter dans la suite les différents travaux réalisés dans chaque contexte.

1.2.1 Planification de la mission

Après avoir donné une idée générale sur les missions dans la section précédente, nous allons présenter dans cette section la manière dont se planifie une mission. Le problème de planification de mission, qui est à la base un problème de décision est celui de la détermination d'un itinéraire dans un environnement qui satisfait toutes les contraintes ainsi que les objectifs de la mission. La complexité de ces problèmes, qui est liée à l'augmentation du nombre d'objectifs à atteindre, aux contraintes temporelles ainsi que des contraintes matérielles, conduit à l'utilisation d'une décomposition hiérarchique pour les rendre gérables [Albus et al., 1988, Chandler et Pachter, 1998]. En effet, la décomposition du problème à des problèmes plus petits ou moins complexes rend sa résolution beaucoup plus facile.

Généralement la planification de la mission est ainsi décomposée en trois niveaux [Kolitz, 1993] :

- Le niveau le plus élevé correspond à la planification des objectifs de la mission, dans lequel, les points de passage, que nous allons appeler les *waypoints* dans la suite, sont définis et ordonnés. Les objectifs sont également définis afin de fixer une première version d'un plan de haut niveau de la mission, en tenant compte des ressources disponibles et en maximisant l'efficacité de ce plan.
- Un niveau intermédiaire peut être défini comme étant la planification de chemin (*path planning*) qui est responsable de l'accomplissement des détails du plan généré au niveau précédent. Plus précisément, une planification de chemin permet de définir les routes que peut emprunter un drone pour relier les waypoints tout en évitant les obstacles et les actions nécessaires pour atteindre les objectifs de la mission.
- Le niveau le plus bas est connu sous le nom de planification de trajectoire. Il s'agit d'affiner les chemins bruts fournis dans le niveau intermédiaire, qui sont définis par des waypoints, afin de générer des trajectoires paramétrées en fonction du temps et les caractéristiques physiques du drone.

La planification d'une mission peut se faire hors ligne, c'est-à-dire avant le début de la mission. Tout le processus s'effectue via des outils spécifiques qui vont générer un plan de mission, qui sera à son tour embarqué dans le drone. Cependant, lors de cette mission un certain nombre d'éléments peuvent être changés, l'apparition des obstacles, des pannes. . . . Dans un contexte de drones autonomes, ceux-ci doivent être capables de replanifier une partie de leur mission en fonction des aléas pouvant survenir. Cette replanification en ligne peut être imposée par le type de la mission, par exemple la recherche d'un objet dans un environnement inconnu ou la poursuite d'une cible mobile.

En résumé, planifier une mission revient à définir les objectifs et à planifier une trajectoire qui permet d'atteindre ces objectifs. Il existe un grand nombre de travaux [Betts, 1998, Gorzen *et al.*, 2009] qui se concentrent sur la planification de trajectoire vu que cette tâche est plus complexe et liée à plusieurs contraintes : les conditions météorologiques, atmosphériques, l'incertitude de certaines informations, . . . Ils peuvent être divisés en trois catégories : (1) approches basées sur l'échantillonnage (2) approches basées sur l'optimisation numérique (3) approches heuristiques. Quelques travaux liés à ces approches seront détaillés dans la suite pour donner une idée des méthodes de résolution.

Dans les approches basées sur l'échantillonnage, le problème de planification de trajectoire se tourne vers la recherche d'une solution réalisable parmi un nombre limité d'alternatives. Selon les principes des algorithmes de recherche, ce type d'approche peut être principalement divisé en deux catégories : les méthodes déterministes et probabilistes. Pour les méthodes déterministes, la zone de vol du drone est discrétisée en un ensemble fini de points dans lequel la trajectoire avec le minimum de temps d'exécution est recherchée par les méthodes de recherche de la racine d'une fonction comme la méthode de bisection [McGee et Hedrick, 2007, Yokoyama et Ochi, 2009]. [Obermeyer *et al.*, 2012] utilisent une variante du problème du voyageur de commerce pour planifier la trajectoire en prenant en considération les impacts du vent. Des efforts ont été consacrés à la planification de la trajectoire des drones par des méthodes probabilistes [LaValle, 1998]. Dans [Frazzoli *et al.*, 2002], un algorithme de recherche basé sur l'arbre aléatoire à exploration rapide (RRT) est appliqué pour la planification de trajectoire en présence d'obstacle fixe et mobile. Une autre méthode basée sur RRT est développée pour éviter les collisions avec

d'autres drones [Yu *et al.*, 2010].

Un problème de planification de trajectoire peut être aussi transformé en un problème d'optimisation qui vise à obtenir une trajectoire optimale par rapport à un objectif spécifique. Les approches d'optimisation dans ce contexte sont principalement concentrées sur la programmation linéaire mixte en nombres entiers (MIP) et la programmation non linéaire (NP). Un problème de planification de trajectoire peut être formulé en un problème d'optimisation représenté par des contraintes linéaires sur un mélange de variables continues et entières [Schrijver, 1998]. Le MIP est aussi utilisé pour replanifier les trajectoires d'un coût de carburant minimal afin d'atténuer la formation persistante des traînées de condensation [Campbell *et al.*, 2013]. Récemment, le MIP est utilisé pour la planification de trajectoire en temps réel tout en évitant les zones radar ainsi que les collisions avec d'autres objets [Israelsen *et al.*, 2014] où la complexité des calculs augmente de façon remarquable avec l'augmentation des obstacles et des contraintes dynamiques. Dans [Borrelli *et al.*, 2006], la planification de trajectoire est basée sur les NP pour éviter les collisions entre drones, le problème NP est formulé avec des contraintes d'états du système et des entrées de contrôle [Jain et Tsiotras, 2008]. Les auteurs de [Ketema et Zhao, 2010] exploitent les NP pour optimiser une trajectoire en présence du vent.

En raison de la complexité des problèmes d'optimisation en nombres entiers ou non linéaires, et des temps de calcul conséquents qui en découlent, des travaux sur l'utilisation des métaheuristiques dans la planification de trajectoire ont montré un intérêt considérable. Ces travaux utilisent principalement les algorithmes génétiques (AG) et l'optimisation par des colonies d'abeilles artificielles. Les algorithmes génétiques représentent une classe de méthodes d'optimisation qui s'inspirent du processus d'évolution naturel [Winter *et al.*, 1996] et qui peuvent être utilisés pour résoudre le problème d'optimisation de la planification de trajectoire. Une fonction est appliquée à chaque solution individuelle afin d'évaluer sa qualité (ou « fitness »). Puis, à partir d'une série d'opérations inspirées de la génétique, une nouvelle population de trajectoires est générée. Dans [Nikolos *et al.*, 2003], un AG modifié est adopté à l'étape de conception d'un planificateur de trajectoire hors ligne/en ligne pour les drones. Plus récemment, en intégrant la dynamique des drones, la trajectoire optimale est obtenue par AG dans [Ergezer et Leblebicioglu, 2013].

Pour pouvoir accomplir ces tâches, le drone a besoin d'avoir conscience de ce qui se passe dans son environnement. Cette capacité est incarnée dans des capteurs dits « intelligents » que nous allons les détailler dans la suite.

1.2.2 Capteurs intelligents

Les capteurs intelligents permettent au drone de « détecter », « voir », « entendre » et « comprendre » le monde qui l'entoure afin qu'il puisse fonctionner intelligemment et prendre des décisions dans un environnement inconnu et en l'absence d'un pilote à bord. Ils sont cruciaux puisqu'ils fournissent les technologies qui feront en sorte que « les drones fonctionnent comme s'il y avait un pilote humain à bord ».

L'une des fonctions de ces capteurs est d'assurer la sécurité du drone en détectant les autres aéronefs autour de lui et de déterminer les risques potentiels de collision. Ces capteurs font partie d'un système plus générique appelé système de « détection et évitement » (D&E) [Yu et Zhang, 2015]. Ce dernier est composé de quatre éléments : la détection matérielle (capteurs), le mécanisme de décision, la planification de trajectoire et le contrôle de vol. Le processus de D&E commence par recueillir les données des capteurs de détection (par exemple, GPS, radar, LL-

DAR (Laser/Light Detection and Ranging), EO (Electro-Optical), IR (Infrared) [Marshall *et al.*, 2016]) qui sont traitées par des algorithmes de détection selon leur source ce qui donne l'aspect intelligent aux capteurs. Par conséquent, le mécanisme décide si la trajectoire doit être replanifiée ou non en fonction des données fournies. Si c'est le cas, le planificateur essaie de trouver la trajectoire optimale ou faisable sous les contraintes de la dynamique du drone et de l'énergie restante pour éviter les collisions comme mentionnées dans la section 1.2.1. En fonction de la trajectoire replanifiée et de la dynamique du drone, le contrôleur de vol délivre des signaux de commande pour que les manœuvres d'évitement puissent être effectuées physiquement.

Après avoir reçu les données traitées des capteurs, c'est le mécanisme de décision qui détermine s'il faut replanifier la trajectoire pour éviter l'obstacle. L'idée préliminaire est de trouver comment reproduire la prise de décision d'un pilote humain et de l'étendre ensuite aux drones. Dans [Narayan *et al.*, 2008], deux modèles de décision sont introduits, le modèle de prise de décision fondé sur la reconnaissance pour représenter la prise de décision intuitive (RPDM) et le modèle de Rasmussen. Le modèle RPDM est en réalité une forme intuitive de diagnostic de prédiction d'un pilote expert qui utilise ses connaissances et son expérience pour structurer un processus de décision sous forme des règles conditionnelles de type « Si... alors... » qui produit une décision finale. Le modèle de Rasmussen utilise en plus des règles conditionnelles données par le pilote sa capacité à prioriser ces règles. Au meilleur de notre connaissance, il y a peu de travaux qui ont été réalisés sur la prise de décision, en particulier pour D&E. Des algorithmes applicables pour d'autres fonctionnalités sont adaptés pour la prise de décision en D&E. Nous pouvons aussi citer le modèle de Markov, déjà introduit en 1.1 pour la planification d'une mission. Le modèle logique de matrice à domaines multiples (MDM) [Mikaelian *et al.*, 2012] est utilisé pour représenter explicitement à la fois la structure et le comportement logique des drones.

Durant la mission, ces capteurs peuvent avoir des défauts et vu que le drone est autonome, il doit être capable de les gérer. Cette gestion est détaillée dans la partie suivante.

1.2.3 Gestion des défauts

Un autre aspect qui joue un rôle important sur l'efficacité des drones autonomes c'est la capacité de gérer les défauts qui apparaissent pendant la mission. Cela a mené à avancer les recherches dans ce domaine et en particulier sur la prise de décision qui est déclenchée par la présence de ces défauts dans certaines situations. Par exemple, nous citons la qualité des données fournies par les capteurs et leur traitement / analyse à bord du drone, le diagnostic des pannes et les algorithmes de pronostic de défaillance, ... [Saha *et al.*, 2011] présentent un nouveau système de gestion d'état de santé des batteries des drones électriques basé sur l'inférence bayésienne. L'objectif est de pouvoir prédire l'événement de fin de décharge qui indique que la batterie est épuisée pour une mission donnée. [Zermani *et al.*, 2016] proposent un module complémentaire de gestion d'état de santé pour surveiller la précision du GPS, ce module s'appuie sur un modèle de réseau bayésien qui permet de renforcer la certitude d'une défaillance du GPS sur la base d'observations. Différents concepts d'intégration de l'évaluation de l'état de santé et gestion de défaillance en temps réel dans les drones sont présentés dans [Roemer et Tang, 2015].

Jusqu'à présent nous avons cité des travaux sur la prise de décision dans les drones dans un cadre de missions individuelles. Dans la partie qui suit, nous présentons des travaux réalisés pour des missions exécutées par de multiples drones.

1.2.4 Décisions distribuées

Certaines missions sont très complexes, et nécessitent l'utilisation de plusieurs drones pour leur accomplissement. Cela génère de nouveaux défis au niveau de la prise de décision. Un grand nombre de recherches ont été menées sur les aspects des décisions coopératives des missions multi-drones et la recherche des solutions aux problèmes causés par cette coopération.

[Yu et Sycara, 2006] traitent le problème de la décision coopérative pour des capteurs distribués et proposent une approche au niveau des capteurs pour déterminer la qualité des informations fournies par les autres capteurs afin de mettre à jour leur fiabilité. Alors que dans [Zhong et Cassandras, 2015] les auteurs considèrent le problème où des drones coopèrent pour optimiser un objectif commun tout en maintenant une communication entre eux pour échanger des informations. La communication peut être coûteuse en termes d'énergie dans un cadre où la quantité d'énergie disponible est limitée pour les drones. Les auteurs proposent donc un système asynchrone de communication basé sur une approche événementielle. [Wan *et al.*, 2014] proposent un algorithme qui permet la localisation coopérative de plusieurs drones dans le cas de perte des signaux GPS.

1.3 Résumé et mise en perspective

Ce chapitre montre que les drones sont utilisés dans différents domaines et qu'il existe plusieurs façons de les classifier et en particulier, la classification par niveau d'autonomie et le degré d'implication d'un pilote humain dans leur prise de décision. Il existe trois niveaux d'autonomie : (1) Pilotage à distance, les drones sont télécommandés par un opérateur au sol. (2) Semi-autonome, les drones reçoivent des ordres de haut niveau et ils sont surveillés par un opérateur. (3) Entièrement autonome, les drones sont capables de d'accomplir une mission sans intervention d'un opérateur humain et prendre de décision de façon autonome lorsqu'ils en ont besoin. Beaucoup de travaux sont menés dans la littérature pour augmenter le niveau d'autonomie des drones et plus précisément la prise de décision. Nous constatons que cette tâche intervient dans différents niveaux de contrôle de drone, commençant par le niveau le plus haut qui traite la planification d'une mission jusqu'au niveau plus bas qui touche aux capteurs et à la gestion de leur défaillance.

De nombreux modèles décisionnels ont été proposés dans la littérature, selon le problème de décision à traiter. Nous remarquons notamment que peu de modèles tentent de reproduire la prise de décision d'un pilote humain et prendre en compte ses préférences.

Pour que l'opérateur ait plus de confiance dans les décisions prises par un drone autonome, le moteur de décisions embarqué dans le drone doit générer des décisions qui sont conformes à ses attentes. Pour cette raison, nous proposons dans ce travail un moteur de décisions multi-niveaux qui permet de représenter la logique adaptée par un pilote lors d'une mission. Ce moteur est composé de deux niveaux principaux et d'un module de calcul de trajectoire : (1) Déclencheur de décision : son rôle est de surveiller l'environnement et de déclencher une prise de décision haut niveau lorsqu'il y en a besoin ; (2) Choix d'action : si la prise de décision est déclenchée par le niveau précédent, des actions de haut niveau (retour à la base, poursuivre mission, ...) sont évaluées sur plusieurs critères. Un modèle de décision est exécuté pour choisir la meilleure parmi un ensemble d'actions possibles. Nous développons les caractéristiques de ce moteur dans le chapitre 3.

Puisque notre principal objectif est de donner à l'opérateur plus de confiance dans les décisions

du drone, le moteur de décisions multi-niveaux que nous proposons est fait de sorte que nous puissions intégrer les préférences de l'opérateur à de multiples endroits. Dans cette thèse nous nous intéressons à l'intégration de modèles d'Aide Multi-Critère à la Décision (AMCD) dans les différents niveaux du moteur proposé. Cela permet de tenir compte des critères multiples liés aux objectifs de la mission, et d'intégrer des modèles des préférences de l'opérateur au sol dans le drone. Il en résultera notamment que deux drones, configurés selon deux opérateurs ayant des préférences et des priorités différentes, adopteront des comportements différents, dans une même situation.

Chapitre 2

Aide Multi-Critère à la Décision

Sommaire

2.1 Concepts généraux	15
2.2 Processus d'aide à la décision	17
2.3 Modélisation des préférences	19
2.3.1 Théorie de la valeur multi-attribut	20
2.3.2 Relations de surclassement	22
2.4 Apprentissage pour la décision multi-critère	25
2.4.1 Élicitation indirecte par lot des paramètres du modèle	26
2.4.2 Élicitation incrémentale	27
2.5 Résumé et mise en perspective	28

Un des objectifs de cette thèse est de proposer un moteur de décisions qui permet de reproduire la prise de décision d'un pilote durant un vol et qui intègre ses préférences. Les modèles de l'Aide Multi-Critère à la Décision (AMCD) permettent de représenter fidèlement les décisions d'un être humain et ses préférences. Nous proposons d'utiliser ces techniques dans le contexte de la prise de décision des drones. Ce chapitre commence par une présentation des concepts généraux d'AMCD ainsi que le processus adapté pour résoudre un problème de décision multi-critère. Nous continuons par présenter deux approches pour modéliser les préférences d'un être humain, les fonctions de valeur et les relations de surclassement. Les modèles d'AMCD nécessitent une phase d'élicitation des préférences qui permet de définir les paramètres d'un modèle. Il existe deux techniques d'élicitation, directe et indirecte. Dans ce chapitre nous nous intéressons surtout aux travaux réalisés pour l'élicitation indirecte et plus précisément à l'élicitation incrémentale pour réduire l'effort cognitif du décideur. Enfin nous présentons le lien entre les techniques d'AMCD et leurs applications dans le contexte des drones autonomes.

2.1 Concepts généraux

La prise de décision fait partie de notre quotidien personnel et professionnel, comme décider du chemin à emprunter pour aller au travail ou encore décider de la stratégie d'investissement. Dans certaines situations ces décisions peuvent être simples ou plus complexes, notamment lorsqu'il y a beaucoup d'alternatives possibles où plusieurs critères interviennent à la prise de décision et qui peuvent être conflictuels. Le conflit entre ces critères survient lorsque l'amélioration d'un

critère ne peut se faire qu'au détriment d'un ou de plusieurs des autres critères, par exemple dans le cas où il faut minimiser le prix tout en maximisant la qualité. Dans un cas concret, le conflit peut impliquer plus que deux critères.

Quand la décision est complexe et son impact est important, il est indispensable de passer par une assistance ou ce que nous appelons *aide à la décision* qui permet d'aider la(es) personne(s) en charge de la décision (communément appelé(s) *décideur(s)*) à prendre la meilleure décision possible.

En général, il y a deux acteurs dans le processus de l'aide à la décision : le décideur et l'analyste. Comme nous l'avons déjà mentionné, le premier est celui qui doit prendre une décision et qui peut être une personne autonome, ou une entité multiple. Le décideur n'est peut-être pas familier avec la méthodologie qui peut être utilisée pour l'aider à prendre une décision. Le deuxième acteur, l'analyste, est donc chargé d'aider le décideur à prendre une décision, en utilisant différentes méthodes et outils formels, mais il faut bien préciser que la décision finale appartient au décideur et doit donc refléter ses préférences.

L'un des principaux défis des problèmes de décision dans le monde réel est la nature multi-dimensionnelle des problèmes de décision. Très souvent, même les décisions paraissent les plus simples et que nous devons prendre régulièrement impliquent des dimensions de préférences multiples.

Exemple *Alice veut acheter un téléphone portable. Elle a le choix entre deux téléphones a et b. Alice considère que dans son choix, deux critères sont importants : le prix (qu'elle veut le plus bas possible), et la performance du téléphone. a est un téléphone cher et très performant, alors que b est peu cher, et n'est pas très performant. Alice rencontre un problème de décision puisqu'elle préfère a parce qu'il est plus performant, mais en même temps elle préfère b parce qu'il est moins cher.*

Le simple exemple illustratif de l'achat d'un téléphone portable montre que les deux critères / dimensions sont conflictuels et que les préférences d'Alice ont une grande influence sur la décision finale. L'approche d'*Aide Multi-Critère à la Décision (AMCD)* [Roy, 1985, Roy, 1996] peut être utilisée dans ce cas. En effet, l'AMCD est une méthodologie pour soutenir les décideurs dans leurs décisions complexes dans lesquelles de multiples critères doivent être pris en compte.

Un problème de décision multi-critère est caractérisé par la prise en compte de plusieurs critères qui servent à refléter toutes les caractéristiques qui sont considérées comme importantes par le décideur lors de la prise de décision. Chaque critère est défini selon une échelle de mesure qui peut être quantitative ou qualitative. Ces critères sont utilisés pour décrire des décisions possibles ou ce que nous appelons alternatives. Formellement une alternative a est définie par un vecteur $a \equiv (a_1, a_2, \dots, a_m)$, $a \in \mathcal{A}$ où m est le nombre de critères, a_j est l'évaluation de l'alternative a sur le critère d'index $j \in \mathcal{M} = \{1, \dots, m\}$. X_j est l'échelle d'évaluation du critère j , $(a_1, a_2, \dots, a_m) \in \mathcal{X} = \prod_{j=1}^m X_j$ et $\mathcal{A} = \{a, b, c, \dots\}$ est l'ensemble des alternatives.

Exemple (suite) *Dans notre exemple d'Alice qui doit choisir un téléphone, les deux téléphones a et b sont l'ensemble d'alternatives $\mathcal{A} = \{a, b\}$ et elle a décidé de les évaluer sur deux critères ($\mathcal{M} = \{1, 2\}$) « performance » ($j = 1$) et « prix » ($j = 2$).*

Chaque critère j induit un pré-ordre total sur \mathcal{A} , noté \succsim_j , qui représente les préférences du décideur par rapport au j^{me} critère à partir d'une relation de préférence : $x, y \in \mathcal{A}$, $x \succsim_j y$ si le décideur estime que x est au moins aussi bon que y sur le critère j .

Exemple (suite) Alice trouve que a est au moins aussi bon que b pour le critère de performances parce qu'il est plus performant d'où $a \succsim_1 b$, d'autre part elle trouve que b est au moins aussi bon que a pour le critère de prix parce qu'il est moins cher ce qui se traduit par $b \succsim_2 a$.

Les problématiques de décision les plus courantes ont été identifiées dans [Roy, 1985] comme étant : le choix, le tri et le classement.

La problématique du choix vise à trouver une ou plusieurs alternatives que nous pouvons considérer comme les meilleures, en fonction des préférences du décideur.

La problématique de tri essaie de placer toutes les alternatives dans des catégories prédéfinies qui sont ordonnées préférentiellement.

La problématique de classement vise à ordonner les alternatives en fonction des préférences du décideur, de manière totale ou partielle ce qui permet de les comparer deux à deux et de choisir la meilleure.

Un grand nombre de méthodes d'AMCD ont été proposées pour traiter ces différentes problématiques. Chaque méthode se distingue, selon le type de problème, par les caractéristiques d'informations sur les préférences admises, les façons de construire et d'exploiter le modèle des préférences et les techniques utilisées pour élaborer une recommandation.

Ces méthodes s'inscrivent dans un processus plus large que nous détaillons ci-après.

2.2 Processus d'aide à la décision

Le processus d'aide à la décision est un support qui accompagne l'analyste et le décideur durant la résolution d'un problème de décision multi-critère. Il est utilisé essentiellement de façon opérationnelle et orienté vers la structuration et la formulation des problèmes et en vue d'obtenir des conclusions utiles, robustes et significatives.

Ce processus est souvent complexe et dépend du domaine d'application et des préférences du décideur. Afin de le simplifier, plusieurs études [Guitouni et Martel, 1998, Ishizaka et Nemery, 2013] ont été menées sur le choix de l'algorithme le mieux adapté à un problème décisionnel, tandis que d'autres comme [Tsoukiàs, 2007, Bouyssou et al., 2006, Bisdorff et al., 2015, Figueira et al., 2005], ont divisé ce processus en plusieurs étapes.

Les étapes suivantes, présentées par [Tsoukiàs, 2007], sont classiquement utilisées pour structurer un processus d'aide à la décision : *identifier le problème, formuler le problème, construire le modèle d'évaluation et ensuite parvenir à une recommandation finale.*

Identification du problème

Identifier un problème de décision est le résultat d'un effort fourni par l'analyste et le décideur pour répondre à des questions du type :

- qui a un problème ?
- pourquoi y a-t-il ce problème ?
- qui décide de ce problème ?
- quel est l'engagement du client sur ce problème ?
-

Cette identification permet au décideur, d'une part, de mieux comprendre sa position dans le processus décisionnel pour lequel il a demandé l'aide à la décision et, d'autre part, de mieux comprendre le rôle de l'analyste dans ce processus décisionnel. En fait, l'exploration de cette étape permet de clarifier les malentendus lors de l'interaction entre le décideur et l'analyste et donc d'améliorer la communication entre ces deux acteurs.

Formulation du problème

Alors que l'étape précédente a un but plus descriptif, l'étape de formulation du problème a un caractère plus constructif. À ce stade, l'analyste doit réduire le vrai problème à un problème formel. Si le modèle est accepté par le décideur, différentes méthodes peuvent être utilisées par la suite pour résoudre le problème de décision.

Lors de cette étape, les alternatives, ou décisions potentielles, doivent être identifiées, ainsi que l'ensemble des points de vue qui ont une influence sur la décision, c'est-à-dire comment les alternatives sont évaluées, et enfin l'objectif, ou ce que le décideur s'attend à trouver à la fin de ce processus.

Plusieurs approches d'aide à la décision s'arrêteront après cette étape, car elles considèrent que comprendre le problème équivaut à le résoudre. D'autres approches tiennent compte de la formulation du problème telle qu'elle est donnée et ne considèrent que les prochaines étapes. Dans tous les cas, l'étape consistant à proposer une formulation de problème est très importante dans le processus d'aide à la décision et elle est utilisée pour construire le modèle d'évaluation.

Construction du modèle d'évaluation

Cette étape représente la partie sur laquelle se concentrent la plupart des approches classiques d'aide à la décision. Une fois que les alternatives et les critères sont définis, différents modèles et approches peuvent être utilisés afin de trouver une solution formelle.

Un modèle, qui est capable de saisir les préférences du décideur sur l'ensemble des alternatives, doit d'abord être sélectionné et ajusté en fonction de ses préférences. Cette étape est très importante, car le résultat final sera fortement reflété par ce choix. Le modèle est ensuite utilisé pour construire une relation globale entre les alternatives. Après, cette relation est exploitée par une méthode choisie afin d'apporter une solution.

Un aspect important lors du choix du modèle d'évaluation est qu'il émerge naturellement de la formulation du problème et qu'il ne soit pas imposé en amont avec une formulation du problème autour de ce choix. Ce choix doit se baser sur les propriétés des modèles possibles et les conséquences de l'application d'un modèle sur le problème. L'analyste doit établir un ensemble de propriétés que le modèle doit remplir et informer le décideur de ses effets secondaires possibles. Dans une telle perspective, l'étude axiomatique des modèles est une connaissance clé de l'analyste puisqu'elle permet d'avoir une cartographie précise des propriétés que chaque modèle satisfait. De plus chaque modèle nécessite l'utilisation d'un certain nombre de paramètres. Certains de ces paramètres représentent directement des informations préférentielles déduites du décideur et de son savoir-faire. D'autres sont des interprétations plus ou moins fidèles de ses connaissances et dépendent du modèle utilisé.

Recommandation finale

La recommandation finale est construite en traduisant la solution formelle de l'étape précédente en une recommandation claire et facile à comprendre par le décideur.

Plusieurs aspects doivent être validés lors de la construction de ce résultat :

- l'analyste doit s'assurer que le modèle est formellement correct ;
- le décideur doit s'assurer qu'il comprend le modèle et qu'il reflète ses préférences ;
- la recommandation doit être acceptée ou non par le décideur et les conséquences de cette décision doivent être bien comprises ;
- la robustesse du modèle, c'est-à-dire sa capacité à réagir face à des situations imprévues, ou des perturbations légères des paramètres préférentiels, ainsi que sa façon de réagir face à de nouvelles situations.

2.3 Modélisation des préférences

Trouver un modèle de préférence et l'adapter de manière à ce qu'il soit capable de saisir le point de vue du décideur est une étape primordiale dans le processus d'aide à la décision.

En 1968, Roy inaugure la branche des méthodes de surclassement [Roy, 1968] alors qu'en 1976, Keeney et Raiffa élargissent la théorie des valeurs pour les problèmes multidimensionnels [Keeney et Raiffa, 1976]. Ces deux courants de pensée distincts donnent deux conceptions méthodologiques différentes connues comme l'école des relations de surclassement, et l'école de la théorie de la valeur multi-attribut.

D'un point de vue méthodologique, il existe d'importantes différences dans les outils générés par les deux tendances. D'une part, l'école européenne s'est développée autour de méthodes discrètes via des relations de surclassement, où la recommandation est construite sur des comparaisons par paires des différentes options. D'autre part, l'école américaine s'est développée autour des méthodes d'utilité et de valeur pour obtenir une comparabilité totale des options.

Avant de les détailler, il est important de préciser qu'il existe différents types de relations de préférence entre deux alternatives [Roy, 1996, Figueira *et al.*, 2010] :

- Indifférence (\sim) : une relation réflexive et symétrique qui apparaît entre deux alternatives considérées comme équivalentes par le décideur ;

- Préférence stricte (\succ) : une relation non réflexive et asymétrique entre deux alternatives où l'une est clairement préférée à l'autre par le décideur ;
- Préférence faible (\succeq) : une relation non réflexive et asymétrique entre une alternative qui est soit strictement préférée, soit indifférente à une autre ;
- Incomparabilité (R) : une relation non réflexive et symétrique entre deux alternatives qui ne peut être placée dans aucune des relations précédentes.

2.3.1 Théorie de la valeur multi-attribut

La théorie de la valeur multi-attribut (MAVT) [Keeney et Raiffa, 1993] est connue comme l'approche « *Agréger puis comparer* ». Elle évite la difficulté du caractère multidimensionnel des alternatives en créant un critère unique qui agrège tous les critères ensemble. De cette façon, toutes les dimensions qui définissent une alternative sont réduites à une seule, le vecteur d'évaluation des alternatives est agrégé en une seule valeur, qui peut aussi être appelée le « score » de l'alternative. Puis elle compare les évaluations scalaires de deux alternatives pour déterminer laquelle est la meilleure.

Cette approche permet de dériver une relation \succeq (voir section précédente) modélisant les préférences du décideur par comparaison des valeurs (performances globales ou satisfaction) des alternatives ; la valeur de chaque alternative est obtenue par application de la fonction de valeur V à son vecteur de performances :

$$x \succeq y \iff V(x) \geq V(y), \quad x, y \in \mathcal{A}$$

et la relation de préférences induite par cette fonction de valeur est un préordre qui permet d'ordonner les alternatives globalement et de choisir la meilleure.

La fonction de valeur additive [Fishburn, 1964, Fishburn, 1965, Fishburn, 1970] est la plus courante et correspond à la somme de toutes les fonctions de valeur marginale v_i :

$$V(x) = \sum_{j \in \mathcal{M}} v_j(x_j), \forall x \in \mathcal{A}$$

La fonction de valeur marginale $v_j : X_j \rightarrow \Re$ détermine le score attribué à une alternative sur le critère j .

Une forme particulière et bien étudiée est *la somme pondérée*, où les paramètres sont sous la forme d'un vecteur de poids strictement positifs $w \equiv (w_1, \dots, w_m)$ tels que $\sum_{j \in \mathcal{M}} w_j = 1$. Les poids sont interprétés ici comme des niveaux de compensation entre les critères où chaque poids w_j permet de quantifier l'importance du critère j pour le décideur. Plus précisément, étant donné un vecteur $w \equiv (w_1, \dots, w_m)$, l'utilité d'une alternative x est définie comme suit :

$$V(x) = \sum_{j \in \mathcal{M}} w_j x_j, \forall x \in \mathcal{A}$$

Par définition, la relation de préférence \succeq induite par la somme pondérée est la suivante :

$$x \succeq y \iff \sum_{j \in \mathcal{M}} w_j x_j \geq \sum_{j \in \mathcal{M}} w_j y_j, \quad x, y \in \mathcal{A}$$

Une autre fonction de valeur additive est aussi utilisée, appelée *la moyenne pondérée ordonnée* (notée OWA pour Ordered Weighted Averaging en anglais) [Yager, 1988]. Plus précisément, pour

définir la valeur d'une alternative au sens de OWA, il s'agit d'appliquer une somme pondérée sur son vecteur de performances après avoir trié ses composantes en ordre croissant. Notons le vecteur trié de $x \equiv (x_{(1)}, \dots, x_{(m)})$ et $w \equiv (w_{(1)}, \dots, w_{(m)})$ le vecteur des poids positifs de somme égale à 1, où le poids $w_{(j)}$ représente l'importance du j^e rang pour le décideur :

$$V(x) = OWA(x, w) = \sum_{j \in \mathcal{M}} w_{(j)} x_{(j)}, \forall x \in \mathcal{A}$$

La relation de préférence \succsim induite par OWA est définie comme suit :

$$x \succsim y \iff \sum_{i \in \mathcal{M}} w_{(i)} x_{(i)} \geq \sum_{j \in \mathcal{M}} w_{(j)} y_{(j)}, \quad x, y \in \mathcal{A}$$

Notons que la forme additive fait une hypothèse importante sur l'indépendance des critères. Si les critères ne sont pas indépendants, nous pouvons utiliser différents modèles, par exemple l'intégrale de Sugeno [Sugeno, 1974] ou l'intégrale de Choquet [Choquet, 1954, Schmeidler, 1986, Grabisch et Labreuche, 2010], qui considèrent également les interactions possibles entre critères. Ces interactions peuvent être positives lorsque, par exemple, l'impact d'un critère est renforcé par celui d'un autre, ou négatives lorsque deux critères sont partiellement redondants.

Par rapport à la fonction additive, l'intégrale Choquet et l'intégrale Sugeno ont plus de paramètres sous la forme de poids, appelés capacités pour la première et des mesures floues pour la seconde.

La capacité de l'intégrale de Choquet est une fonction v définie sur l'ensemble des parties de \mathcal{M} :

$$v : 2^{\mathcal{M}} \rightarrow [0, 1] \text{ avec } v(\emptyset) = 0, v(\mathcal{M}) = 1$$

La valeur d'une alternative $x \in \mathcal{A}$ est définie par :

$$Ch(x, v) = \sum_{j \in \mathcal{M}} [x_{(j)} - x_{(j-1)}] v(X_{(j)})$$

avec $X_{(j)}$ l' j^e ensemble de niveaux d'une alternative x donnée et qui est définie par : $X_{(j)} = \{(j), \dots, (m)\}$

Nous pouvons définir la relation de préférence \succsim induite par l'intégrale de Choquet de la manière suivante :

$$x \succsim y \iff \sum_{j \in \mathcal{M}} [x_{(j)} - x_{(j-1)}] v(X_{(j)}) \geq \sum_{j \in \mathcal{M}} [y_{(j)} - y_{(j-1)}] v(Y_{(j)}), \forall x, y \in \mathcal{A}$$

Différents problèmes se posent lorsque nous modélisons les préférences d'un décideur à l'aide de fonctions de valeur. Le premier problème réside dans le choix de la forme de la fonction de valeur parmi ce que nous venons de citer. L'élicitation des paramètres de la fonction choisie est aussi un problème vu que c'est une tâche difficile. Nous verrons dans la suite de ce chapitre qu'il y a différentes façons d'obtenir ces paramètres à partir des préférences d'un décideur. Une autre question tourne autour de la conversion des échelles qualitatives en échelles quantitatives [Belton, 1999]. Cette conversion est nécessaire en raison de la manière dont les fonctions de valeur sont définies, mais c'est un problème difficile vu qu'il faut garder toute l'information contenue dans l'échelle qualitative. Par exemple, transformer l'échelle bon, moyen, mauvais en 3, 2, 1 peut

imposer qu'une bonne évaluation soit considérée trois fois mieux qu'une mauvaise évaluation, ce qui pourrait ne pas être le cas du point de vue du décideur. Pour cette raison, un processus de questionnement rigoureux est obligatoire pour avoir une représentation fidèle à la perception du décideur .

L'un des avantages de représenter les préférences du décideur de cette façon est le fait qu'elle produit un pré-ordre total sur l'ensemble des alternatives. Cela nous permet de comparer les alternatives avec facilité. On pourra toujours construire soit une relation de préférence (\succ) entre deux alternatives, si l'une a un score plus élevé que l'autre, soit une relation d'indifférence (\sim), si leurs scores sont égaux, mais jamais une relation d'incomparabilité (\mathbf{R}).

2.3.2 Relations de surclassement

Contrairement à l'approche précédente qui agrège puis compare, dans l'approche de surclassement il s'agit d'abord de comparer des paires d'alternatives critère par critère puis d'agréger ces comparaisons locales.

Une alternative x surclasse une autre alternative y si [Roy, 1991] :

1. il existe une majorité de critères sur lesquels x est au moins aussi bon que y ;
2. il n'y a pas de critère pour lequel y est largement plus performant que x .

Le premier point est modélisé par des degrés de concordance locale, qui mesurent pour chaque critère j si l'évaluation de x sur j est au moins aussi bonne que celle de y . Généralement, ces degrés sont évalués sur une échelle de 0 à 1. Si x n'est pas au moins aussi bonne que y sur le critère j , alors le degré de concordance locale est 0, si x est au moins aussi bonne que y , ce degré est 1. Si une incertitude est mesurée pour les deux points, une valeur entre 0 et 1 peut être utilisée selon le type de relation de surclassement qui est employé.

Les critères sur lesquels x a une évaluation au moins aussi bonne que y sont pondérés par un ensemble de poids représentant l'importance de chaque critère et cela permet de construire un degré de concordance global. Les poids ont, dans ce cas, une signification différente de celle des poids utilisés dans les fonctions de valeur. Comme les méthodes de surclassement s'inspirent des procédures de vote dans la théorie du choix social [Tsoukiàs, 2006], chaque critère peut être considéré comme un groupe d'électeurs ayant la même opinion, et donc le poids qui leur est attribué représente leur force. La condition de l'indépendance des critères est également requise dans ce cas des relations de surclassement.

Contrairement à MAVT, le traitement des échelles qualitatives et quantitatives est beaucoup plus simple dans ce cas. Il n'est plus nécessaire de convertir une échelle qualitative en une échelle numérique [Figueira *et al.*, 2010], mais il faut seulement définir un pré-ordre sur l'échelle originale pour dire si une évaluation est au moins aussi bien qu'un autre ou pas. Afin de faire face à l'imprécision potentielle des évaluations des alternatives sur l'ensemble des critères, certaines de ces relations prennent en compte des seuils de discrimination. Dans la plupart des cas, deux seuils de ce type sont définis pour chaque critère : un seuil d'indifférence et un seuil de préférence [Roy et Vincke, 1987], qui sont utilisés pour définir les intervalles à l'intérieur desquels une différence entre les évaluations peut être considérée comme négligeable, ou clairement significative. Les seuils de discrimination sont également utilisés pour tenir compte de la perception subjective du décideur à l'égard des évaluations des critères. Revenons à l'exemple d'Alice en guise d'illustration.

tion :

Exemple (suite) Nous considérons le problème de l'achat d'un téléphone en nous concentrant à parts égales sur deux critères, le prix et la performance. Si le téléphone *a* est plus cher, mais a une performance supérieure à celle de *b*, alors Alice se trouve face à un dilemme. Si le téléphone *a* est un petit peu plus cher et qu'Alice considère cette différence négligeable, alors le dilemme initial est éliminé en se basant sur un seuil d'indifférence (exprimé par « considérer la différence comme négligeable »). Le téléphone *a* (un peu plus cher) sera recommandé à Alice parce qu'il est plus performant.

La deuxième partie de la définition de la relation de surclassement peut être facilement vérifiée en déterminant si, pour un critère $j \in \mathcal{M}$, x a une performance beaucoup plus faible que y . Dans certains cas, si une telle situation se produit, les jugements qui peuvent être extraits du degré de concordance globale sont invalidés et x ne surclasse pas y . Pour cette raison, cette partie de la définition de la relation de surclassement est également appelée un *veto* [Rogers et Bruen, 1998]. Pour d'autres modèles, la présence d'une telle situation a pour effet d'affaiblir les jugements portés par le degré de concordance globale. Dans le cas où il n'y a pas de situation de veto, entre x et y , la relation de surclassement est simplement extraite du degré de concordance global. Habituellement, si le degré de concordance globale est supérieur à 0,5, qui est le niveau médian, alors x surclasse y , bien que des seuils plus élevés puissent être requis, lorsque les poids des critères ne reflètent pas précisément la perspective du décideur. Ces seuils sont également appelés *niveaux de coupe* [Roy et Bouyssou, 1993, Yu, 1992].

Une autre propriété qui distingue les relations de surclassement des fonctions de valeurs réside dans le fait que toutes les relations binaires mentionnées en section 2.3.2, à savoir l'indifférence, la préférence faible, la préférence stricte et l'incomparabilité, peuvent être construites à partir de ces relations. Deux alternatives x et y sont indifférentes si x surclasse y , et y surclasse x . Une alternative, x , est dite strictement préférée à une autre y , lorsque x surclasse y et y ne surclasse pas x . Ces alternatives sont incomparables quand aucun d'eux ne surclasse l'autre [Roubens et Vincke, 1985].

L'ajout de la situation d'incomparabilité est directement généré par la deuxième partie de la définition d'une relation de surclassement et ne peut se manifester que dans ce contexte. Cette situation, selon Roy [Roy, 1990], représente les hésitations du décideur, qui peuvent résulter d'incertitudes, de conflits et/ou de contradictions. Généralement, cette situation se produit lorsque deux alternatives présentent des avantages très contrastés. Cela signifie que chaque alternative a une très bonne performance sur au moins un critère par rapport à l'autre.

La construction d'une relation de surclassement est généralement considérée comme plus simple que celle d'une fonction de valeur, même si elle est plus difficile à exploiter. Cela s'explique par le fait que les relations de surclassement ne peuvent pas produire un ordre total entre les alternatives, vu qu'elles sont ni complètes ni transitives [Roy, 1990]. Dans ce cas, l'élaboration d'une recommandation finale est beaucoup plus difficile et nécessite une phase de post-traitement de la relation de surclassement, souvent appelée exploitation.

Il existe plusieurs méthodes basées sur la relation de surclassement qui sont utilisées pour les différents types de problèmes multi-critère (choix, tri et classement) [Figueira et al., 2005].

Parmi les premières méthodes utilisées pour résoudre un problème de choix dans le paradigme du surclassement nous trouvons la méthode ELECTRE I [Roy, 1968] qui appartient à la famille des méthodes ELECTRE. ELECTRE I construit une relation de surclassement sur des alternatives qui sont évaluées sur des critères qui peuvent être hétérogènes : qualitatives et/ou quantitatives. L'exploitation de cette relation consiste à trouver un noyau dans le graphe dérivé de cette relation, qui est alors recommandé comme l'ensemble des alternatives les plus susceptibles d'être choisies. Plus précisément, un noyau d'une relation de surclassement est un sous-ensemble d'alternatives où toute alternative qui n'est pas dans le sous-ensemble est surclassée par au moins une alternative du sous-ensemble et dont les alternatives du sous-ensemble sont incomparables. Un noyau n'est unique que dans le cas où aucun cycle n'existe dans ce graphe. Si ce n'est pas le cas, les cycles maximaux sont réduits à des éléments isolés avant l'étape d'exploitation principale. La méthode ELECTRE IS [Roy et Skalka, 1984] ajoute l'utilisation des seuils d'indifférence et de préférence pour chaque critère, afin de tenir compte de l'imprécision dans l'évaluation des alternatives. Une extension de cette méthode, ELECTRE Iv, est proposée dans [Maystre *et al.*, 1994] qui ajoute l'utilisation de seuils de veto afin de tenir compte des grandes différences sur chacune des échelles de critères. Une autre approche qui trouve une meilleure recommandation de choix est la méthode RUBIS [Bisdorff *et al.*, 2008] qui utilise également la notion de noyau, mais elle l'extrait d'un graphique orienté pondéré induit par une relation de surclassement bipolaire.

Pour les problèmes de tri la méthode la plus courante, qui utilise une relation de surclassement, est ELECTRE TRI [Yu, 1992, Roy et Bouyssou, 1993]. Cette méthode affecte les alternatives à des catégories ordonnées définies par des profils de délimitation en utilisant deux règles : les procédures d'affectation pessimiste et optimiste. Dans les deux cas, l'affectation est basée sur les relations de surclassement entre les alternatives et les profils des catégories. [Figueira *et al.*, 2004] proposent une approche étendue de PROMETHEE, appelée PROMETHEE TRI pour traiter les problèmes de tri. Une autre méthode de tri MR-Sort (Majority Rule Sorting) caractérisée par [Bouyssou et Marchant, 2007a, Bouyssou et Marchant, 2007b] permet aussi d'affecter des alternatives à des catégories en les comparant aux profils qui délimitent les catégories, mais de façon plus simple en utilisant la règle de majorité. D'autres méthodes de tri qui utilisent la relation de surclassement peuvent être trouvées dans [Moscarola et Roy, 1977, Massaglia et Ostanello, 1991].

Des méthodes de la famille ELECTRE sont aussi utilisées pour les problèmes de classement comme par exemple la méthode ELECTRE II [Roy, 1971]. Cette méthode utilise deux relations de surclassement, une forte et une faible, où la première est utilisée pour partitionner l'ensemble des alternatives de sorte que les cycles maximaux qui émergent de cette relation soient considérés comme une seule entité. Deux ordres partiels sont alors générés à partir de cette partition. Le premier est défini par un ordre initial en supprimant successivement les éléments de partition initiale qui ne sont pas préférés par les autres, puis ce résultat est affiné à l'aide de la faible relation de surclassement. De même, le deuxième ordre partiel est généré avec la différence qui est d'abord construite en supprimant successivement les éléments de la partition initiale qui ne sont pas préférés à d'autres. Ces deux ordres partiels sont ensuite utilisés pour construire le dernier. La méthode ELECTRE III [Roy, 1978] améliore ELECTRE II en permettant de traiter des données inexactes, imprécises, incertaines ou mal déterminées. Elle construit également deux pré-ordres complets par des processus appelés distillations, d'une manière similaire à la méthode ELECTRE II, qui sont ensuite fusionnés en un seul pré-ordre complet. Une autre méthode ELECTRE, connue sous le nom d'ELECTRE IV, est née d'un nouveau problème réel lié au réseau du métro parisien [Hugonnard et Roy, 1982]. Il est désormais possible de classer les alternatives sans utiliser les coefficients d'importance relative des critères ; c'est la seule méthode

ELECTRE qui n'utilise pas de tels coefficients. Une autre famille de méthodes est aussi utilisée pour les problèmes de classement, la famille PROMETHEE. Elles construisent généralement deux indices pour chaque alternative x , le premier quantifie le nombre d'alternatives préférées par rapport à x appelé le flux négatif de surclassement alors que le deuxième quantifie le nombre d'alternatives où x est préférée appelé le flux positif de surclassement. En utilisant ces indices, la méthode PROMETHEE I [Brans, 1982] construit un ordre partiel, en plaçant une alternative au-dessus d'une autre si les deux flux de la première alternative sont meilleurs que ceux de la seconde, ou si un seul flux est meilleur et l'autre égal à celui de la seconde alternative. La méthode PROMETHEE II consiste à construire un ordre complet sur l'ensemble des alternatives en construisant un flux net de surclassement à partir de la différence entre les flux de surclassement positif et négatif, donnant ainsi un score unique à chaque alternative. Les autres méthodes de la même famille sont PROMETHEE III (classement par intervalles) et PROMETHEE IV (cas continu) qui sont plus détaillées dans [Brans et Mareschal, 1992, Brans et Mareschal, 1994]. Une nouvelle famille de méthodes appelée RMP (Ranking with Multiple reference Profiles) a été introduite par [Rolland, 2013a] pour des modèles purement qualitatifs qui ne nécessitent aucune représentation numérique. Dans cette famille les alternatives ne sont pas comparées directement entre elles, mais à travers un ou plusieurs profils de références et cette comparaison est utilisée pour donner un préordre à l'ensemble des alternatives.

2.4 Apprentissage pour la décision multi-critère

Dans la section précédente, nous avons présenté deux approches, MAVT et le *surclassement*, pour modéliser les préférences d'individus / décideurs dans le contexte de décision multi-critère. Les paramètres du modèle peuvent être sous la forme des poids des critères et les fonctions de valeur pour un modèle MAVT ou les poids et les seuils de discrimination pour un modèle de surclassement. Ils sont utilisés pour pouvoir représenter au mieux les préférences d'un décideur dans un contexte décisionnel donné.

L'élicitation des préférences est un processus interactif entre l'analyste et le décideur, le premier recueillant les informations sur les préférences du dernier, et permettant de les transformer en valeurs ou en contraintes pour les paramètres du modèle AMCD en question [Mousseau, 2005].

Ces paramètres peuvent être donnés directement par le décideur par le biais d'une approche d'élicitation des préférences dite *directe*, mais une telle approche est généralement trop difficile dans des problèmes réels, car le décideur doit avoir une très bonne compréhension du modèle d'AMCD et doit donner des informations précises qui peuvent prendre les formes suivantes :

- une valeur exacte pour un paramètre (par exemple, le poids du critère i est exactement 0.5) ;
- un intervalle pour un paramètre (par exemple, le poids du critère i doit être compris entre 0.2 et 0.3) ;
- un ratio ou un compromis entre deux pondérations de critères (par exemple, le critère i est deux fois plus important que le critère j) ;
- une importance relative entre deux critères ou deux coalitions de critères (par exemple, les critères i et j sont plus importants ensemble que le critère k) ;
-
-

Par conséquent, dans beaucoup de cas pratiques une approche indirecte est utilisée qui consiste à éliciter les paramètres du modèle à partir d'une connaissance partielle et de préférences holistiques (c.-à-d. des préférences globales plutôt qu'une analyse critère par critère). Ces préférences peuvent être sous la forme d'un classement d'une partie des alternatives, des comparaisons d'alternatives ou, dans le cas de problèmes de tri, des exemples d'affectation. Ils sont, en général, issus d'un petit sous-ensemble d'un ensemble beaucoup plus large d'alternatives à évaluer, ou des alternatives considérées dans des processus décisionnels passés ou même des exemples construits de manière à faciliter les comparaisons.

2.4.1 Élicitation indirecte par lot des paramètres du modèle

Dans la littérature plusieurs recherches ont été menées dont le but est de développer des méthodes pour l'élicitation indirecte des paramètres d'un modèle de préférences pour les deux écoles : MAVT et surclassement. L'approche de désagrégation est généralement effectuée à l'aide de programmes mathématiques. Ces programmes d'inférence peuvent être partiels si seul un sous-ensemble de paramètres est déduit (les valeurs des autres paramètres étant fixes), ou globaux si tous les paramètres doivent être déduits. Les entrées pour le programme mathématique sont les préférences holistiques et les valeurs des paramètres non élicités qui sont traduits à des contraintes. Les variables sont les paramètres du modèle à trouver à partir de la minimisation ou la maximisation d'une fonction objectif. Dans la plupart des cas, il s'agit d'une élicitation par lot (terme issu du domaine de l'apprentissage machine), c'est-à-dire qu'elle se passe en une étape, à partir d'un ensemble d'informations préférentielles données en entrée, de manière non incrémentale.

Pour l'approche MAVT, la méthode UTA [Jacquet-Lagrez et Siskos, 1982, Siskos *et al.*, 2005] permet d'apprendre une fonction de valeur additive par programmation linéaire à partir d'un préordre des alternatives donné par le décideur. Ces préférences avec les propriétés des fonctions de valeur marginale, monotones et par morceaux, sont traduites en des contraintes pour le programme linéaire avec une fonction objectif qui minimise les erreurs de valeur. De nombreuses méthodes en découlent, par exemple UTA* [Siskos et Yanacopoulos, 1985] qui minimise plutôt deux variables d'erreur, la surestimation et la sous-estimation de la valeur. UTADIS [Zopounidis et Doumpos, 1999] est une variante de UTA utilisée pour les problèmes de tri, elle estime les fonctions de valeur et les profils en minimisant les erreurs de classification entre les classes prédéfinies. Les méthodes UTAMP (UTAMP1, UTAMP2) [Beuthe et Scannella, 2001], sur base d'UTA, permettent de maximiser les écarts de performance entre des solutions dont le classement relatif est connu.

D'autres méthodes sont utilisées pour éliciter les paramètres de l'intégrale de Choquet en minimisant un critère d'erreur quadratique [Mori, 1989, Meyer et Roubens, 2006] ou en minimisant la variance du modèle pour être le moins spécifique possible [Kojadinovic, 2007]. D'autres modèles non additifs ont été utilisés dans [Angilella *et al.*, 2010] qui prennent comme entrée, en plus des préférences holistiques du décideur (les comparaisons par paires des alternatives), des informations reliées à l'importance des critères, le signe et l'intensité des interactions entre paires de critères.

Pour les approches de surclassement, l'élicitation des préférences indirecte pour des méthodes de choix et de classement, telles que ELECTRE I - IV n'est pas facile. [Dias et Mousseau, 2018] Cela est dû au fait que le résultat de la méthode n'est pas directement lié à la relation de sur-

classement, mais nécessite une phase d'exploitation complémentaire. Par conséquent, une grande partie de la littérature s'est concentrée sur les problèmes de tri et l'élicitation des paramètres de méthodes du type ELECTRE TRI. Des approches interactives pour déterminer les poids des critères à partir de l'affectation des alternatives à des catégories prédéfinies sont proposées dans [Mousseau et Slowinski, 1998, Mousseau *et al.*, 2001], qui utilisent un programme linéaire mixte en nombres entiers (MIP). D'autres sont proposées pour déterminer les seuils de veto dans [Rocha et Dias, 2008]. [Leroy *et al.*, 2011] proposent un MIP afin d'apprendre les paramètres de MR-Sort à partir d'exemples d'affectation. Les résultats expérimentaux de ce travail montrent que le MIP est capable de trouver des modèles MR-SORT qui fonctionnent bien en généralisation. Cependant, les résultats montrent aussi la limitation d'un tel algorithme en termes de temps de calcul. [Sobrie *et al.*, 2013] proposent une métaheuristique pour apprendre les paramètres du modèle de tri, qui est ensuite adapté et étendu par [Olteanu et Meyer, 2014] afin de prendre en compte les seuils de veto en plus. D'autres métaheursitiques évolutionnaires sont proposées pour les modèles MR-Sort avec veto par [Sobrie *et al.*, 2017].

Des travaux récents proposent une approche d'élicitation indirecte pour un modèle SRMP (Simple Ranking Method using Reference Profiles) [Rolland, 2013a]. SRMP est une version simplifiée des modèles RMP, où le décideur exprime ses préférences par des comparaisons par paires d'alternatives qui seront utilisées pour déduire les paramètres du modèle (par exemple, les poids des critères, les profils et leur ordre lexicographique). Dans [Olteanu *et al.*, 2018], les auteurs formulent l'élicitation des préférences SRMP comme un MIP défini par un ensemble de contraintes linéaires liées à l'ensemble des alternatives, les critères, le nombre de profils et la comparaison binaire (préférence / indifférence) des alternatives fournies par le décideur. [Belahcene *et al.*, 2018] proposent une autre approche utilisant un problème de satisfaisabilité booléenne (SAT) qui consiste en un ensemble de variables booléennes et de propositions logiques sur ces variables. Par rapport à la proposition précédente, cette approche est plus rapide et peut traiter des ensembles plus importants de comparaisons d'alternatives par paires. [Liu *et al.*, 2014] proposent une métaheuristique pour obtenir les paramètres d'un modèle SRMP qui est plus rapide que l'approche MIP, mais elle ne garantit pas le *meilleur modèle*.

Pour toutes ces méthodes d'élicitation d'indirecte, il faut solliciter directement les décideurs en leur posant des questions sur leurs préférences. Dans l'optique de limiter le nombre de questions nécessaires à la prise de décision, des procédures d'élicitation partielle ont été proposées. Elles permettent de réduire l'effort d'élicitation en posant progressivement des questions au décideur, soigneusement choisies les unes après les autres, de sorte à restreindre l'espace de paramètres possible jusqu'à être en mesure de formuler une recommandation. Cette approche dite « incrémentale » est inspirée par le domaine d'apprentissage automatique (machine learning).

2.4.2 Élicitation incrémentale

Classiquement, les méthodes d'apprentissage automatique (ou machine learning en anglais) sont définies comme des algorithmes qui déduisent des informations structurelles à partir de données fournies. Certaines de ces techniques se basent sur un apprentissage par lot, où les données sont fournies avant l'apprentissage comme les travaux présentés juste avant. Cependant, dans de nombreuses applications, les données arrivent par un flux constant, ce qui nécessite une adaptation continue du modèle.

L'*apprentissage actif* se concentre sur l'apprentissage dans un contexte où il y a un flux de données et un oracle est utilisé pendant le processus. Contrairement au cadre traditionnel où les

données sont connues et imposées, dans l'apprentissage actif, c'est l'algorithme d'apprentissage qui demande des informations pour des données spécifiques. Plus précisément, ces algorithmes reçoivent les données d'apprentissage séquentiellement, une par une ou morceau par morceau, et les utilisent avec le modèle précédemment appris et l'entrée de l'oracle pour en produire un meilleur.

Les algorithmes d'apprentissage actif se présentent sous trois formes dans la littérature : génération d'instances, algorithmes d'apprentissage en ligne et hors ligne. Dans le premier type, les nouvelles instances à étiqueter sont générées par l'algorithme, alors que dans le second, les données d'apprentissage arrivent séquentiellement, et l'algorithme d'apprentissage décide s'il est nécessaire ou pas d'être confronté à l'oracle. Les algorithmes d'apprentissage hors ligne supposent qu'il existe une base de données d'instances, et l'algorithme sélectionne quelles données sont confrontées à l'oracle. Quelques exemples de l'utilisation de techniques d'apprentissage actif peuvent être trouvés pour la construction incrémentale d'arbres de décision pour la classification ou la régression [Fan *et al.*, 2015, Gholipour *et al.*, 2013, Salperwyck et Lemaire, 2013] et plusieurs modèles SVM incrémentaux sont présentés dans [Wen et Lu, 2007]. [Syed *et al.*, 2017] présentent une approche pour l'apprentissage actif dans les réseaux neuronaux convolutifs.

Dans notre contexte, où l'algorithme de décision est un algorithme d'AMCD, le sujet de l'apprentissage actif a surtout été abordé du point de vue hors ligne, et s'appelle habituellement *élicitation incrémentale des préférences*.

[Durbach, 2009] et [Lahdelma *et al.*, 1998] utilisent un indice qui quantifie le volume du polyèdre des contraintes qui spécifie les fonctions de valeur possibles et tentent de réduire ce volume en ajoutant des contraintes représentant des comparaisons par paires des alternatives, jusqu'à converger vers la meilleure solution. [Holloway et White III, 2003] montrent l'importance de l'ordre des comparaisons par paires pour diminuer le nombre de questions afin de réduire l'effort cognitif du décideur.

[Ciomek *et al.*, 2017] présente un ensemble d'heuristiques pour minimiser le nombre de questions à poser au décideur et les classer par ordre de priorité dans le contexte des problèmes de décision de *choix*. Ils concluent que l'heuristique la plus performante dépend des paramètres du problème (par exemple le nombre de critères et d'alternatives).

Dans le même contexte [Benabbou *et al.*, 2017] sélectionnent une série de questions par paire en utilisant une stratégie de regret minimal. Cette stratégie réduit le nombre de questions par paire, mais la garantie de performance est affaiblie (avec quelques limites acceptables à la situation idéale). En outre, [Olteanu, 2018] présente, dans le contexte des relations de surclassement, des stratégies pour générer des alternatives pour l'élicitation incrémentale de modèles MR-Sort.

2.5 Résumé et mise en perspective

Dans ce chapitre, nous avons présenté une méthodologie, l'AMCD, et ses concepts généraux pour soutenir les décideurs dans leurs décisions complexes où il y a de multiples critères qui interviennent. De nombreux modèles décisionnels multi-critère ont été proposés dans la littérature. La diversité de ces modèles s'explique par la difficulté de construction d'une modélisation parfaite des préférences d'un décideur, mais le choix d'un modèle devrait se déduire naturellement de la formulation du problème de décision. Il existe deux écoles de pensée : (1) la théorie de la valeur multi-attribut, qui est une approche qui consiste essentiellement à attribuer une évaluation

globale à chaque alternative en utilisant une fonction d'agrégation multi-critère pour déterminer quelle est la meilleure alternative, (2) les relations de surclassement dont le but est d'abord de comparer les alternatives deux à deux, critère par critère. Ensuite, ces concordances locales sont agrégées avec une fonction d'agrégation pour finalement déduire si une alternative est au moins aussi bonne qu'une autre.

La collecte des informations sur les préférences du décideur, appelée élicitation des préférences, a comme objectif de calibrer le modèle d'AMCD, afin qu'il fournisse des recommandations de décision compatibles avec les attentes du décideur. Cette dernière est une problématique de recherche visant à concevoir des méthodes qui, en recueillant des informations sur les préférences du décideur, sont capables de formuler une recommandation personnalisée. Deux approches peuvent être utilisées pour l'élicitation des préférences : directe et indirecte. L'approche de l'élicitation directe exige que le décideur donne explicitement des valeurs numériques pour les paramètres du modèle, tandis que l'approche indirecte utilise les informations holistiques fournies par le décideur afin de déduire les paramètres du modèle.

Cependant, l'approche directe d'élicitation est difficile à mettre en œuvre en présence de préférences complexes. C'est pour cette raison que l'approche indirecte intéresse plus les chercheurs du domaine. Classiquement, cette approche permet de déterminer les paramètres du modèle en une seule itération. Toutes les informations liées aux préférences du décideur sont alors données en une fois comme entrée d'un algorithme d'apprentissage. Afin de réduire l'effort cognitif du décideur, une élicitation incrémentale est également proposée. Elle permet de poser progressivement des questions au décideur dans l'optique de limiter le nombre de questions nécessaires pour modéliser ses préférences.

Comme présenté dans le chapitre 1, tout au long de la mission, les drones autonomes sont confrontés à des choix multiples et doivent prendre de nombreuses décisions de manière autonome. Cette tâche de prise de décision nécessite de prendre en compte des critères multiples et potentiellement conflictuels afin d'atteindre les objectifs de la mission. D'autre part, l'opérateur a besoin d'avoir confiance dans les décisions prises par le drone. Pour cette raison, dans cette thèse nous proposons d'intégrer des modèles d'AMCD dans le moteur de décisions multi-niveaux. Les modèles d'AMCD permettent de répondre à deux défis liés aux décisions des drones autonomes : tenir compte des critères multiples et intégrer les préférences de l'opérateur dans le comportement du drone.

Nous montrons dans la partie suivante qu'en suivant le processus d'aide à la décision détaillé en section 2.2, nous avons pu bien formuler notre problème et identifier les classes de modèles d'AMCD à intégrer dans les différentes couches du moteur de décisions du drone. Nous nous tournerons vers des modèles de surclassement pour deux raisons principales : l'hétérogénéité des échelles des critères d'évaluation (données qualitatives et quantitatives), et le besoin d'un modèle explicable pour garantir une confiance élevée dans les décisions du drone.

Deuxième partie

Intégration des préférences d'un opérateur dans les décisions d'un drone autonome

Chapitre 3

Moteur de décisions multi-niveaux et multi-critère pour un drone autonome

Sommaire

3.1	Processus de prise de décision d'un pilote humain	34
3.2	Moteur de décisions multi-niveaux	35
3.3	Intégration des préférences d'un pilote	38
3.3.1	Intégration des préférences dans le niveau 1	38
3.3.2	Intégration des préférences dans le niveau 2	44
3.3.3	Intégration des préférences dans le calcul de trajectoire	49
3.4	Résumé et mise en perspective	50

Le but de ce chapitre est de proposer un moteur de décisions à embarquer dans un drone autonome. Pour que l'opérateur du drone au sol ait une confiance accrue dans les décisions prises par le drone, nous proposons que ce dernier se comporte conformément à ses priorités et ses préférences. Pour cette raison, ce moteur de décisions est basé sur la logique adoptée par un pilote lors d'une mission. Ce chapitre commence par décrire le processus de prise de décision d'un pilote humain dans un environnement aéronautique. Nous proposons ensuite, un moteur de décisions multi-niveaux qui s'en inspire, et imite par conséquent le processus de décision d'un pilote. Il est composé de deux niveaux de décisions principaux et d'un module de calcul de trajectoire. Le premier niveau permet de surveiller l'état du drone et son environnement pour détecter les événements qui peuvent perturber la réalisation de la mission et déclencher la prise de décision du niveau 2 si nécessaire. Le deuxième niveau une fois déclenché permet de choisir une action de haut niveau (atterrir, continuer, ...) la mieux adaptée à la situation courante parmi un ensemble d'actions possibles. Une particularité de ce moteur de décisions est qu'il permet de facilement intégrer les préférences d'un pilote, en utilisant des techniques d'AMCD, dans ses différents niveaux. Nous argumentons dans ce chapitre les modèles possibles, et nous illustrons leur fonctionnement dans le contexte du drone pour certains d'entre eux.

Notez que les travaux présentés dans ce chapitre s'inspirent de l'article [Khannoussi *et al.*, 2019] dont une version préliminaire a également fait l'objet d'une présentation à la conférence ROADEF [Khannoussi *et al.*, 2018a].

3.1 Processus de prise de décision d'un pilote humain

En général, la prise de décision est le processus cognitif qui consiste à choisir une ou une suite d'alternatives parmi un ensemble en se basant sur des critères ou des stratégies donnés [Wang et Ruhe, 2007]. Ce choix détermine le comportement du décideur et il est guidé par les priorités et les préférences du décideur et il a donc une influence sur l'exécution d'une tâche.

La prise de décision dans un environnement aéronautique inclut toute décision pertinente qu'un pilote doit prendre au cours d'un vol. Elle comprend à la fois les décisions d'exécuter ou de ne pas exécuter la mission (avant le vol), et celles prises pendant le vol. Dans notre travail, nous nous intéressons au deuxième cas, où la mission est démarrée, et où des décisions doivent être prises pendant le vol, au cours de la mission. En aéronautique, la prise de décision a une importance particulière en raison des impacts sur la sécurité causés par de mauvaises décisions. L'administration fédérale de l'aviation des États-Unis définit la prise de décisions aéronautiques comme suit [FAA, 1991] :

« la prise de décision aéronautique est une approche systématique du processus mental utilisé par les pilotes d'avion pour déterminer de façon constante la meilleure ligne de conduite à adopter en fonction d'un ensemble donné de circonstances. »

Pour décrire le processus de prise de décision d'un pilote, nous nous sommes inspirés du cycle de Boyd ou de la boucle OODA : « Observe, Orient, Decide and Act » (Observer, Orienter, Décider et Agir) [Fadok, 1995]. Ce cycle, décrit par le pilote John Boyd, représente le processus de décision d'un humain qui doit prendre des décisions dans un environnement aéronautique avec des conditions extérieures changeantes.

La boucle OODA a été développée à l'origine dans le but d'expliquer pourquoi les pilotes de chasse américains avaient plus de succès que leurs adversaires. Comme son nom l'indique, la première activité, *Observer*, consiste à prendre note de certaines caractéristiques de l'environnement. Dans la version originale de la boucle OODA, cela signifiait détecter un avion ennemi. La deuxième activité, *Orienter*, consiste à pointer (orienter) son avion vers l'adversaire, afin d'être en bonne position pour entrer dans la troisième étape, l'étape *Décider*, qui consiste à décider que faire ensuite. Cela amène à la quatrième étape, *Agir*, qui consiste à mettre en œuvre ce qui a été décidé. À la suite de l'étape *Agir*, une nouvelle observation est faite, et ainsi de suite.

Le pilote est en phase d'observation en permanence, il doit surveiller les systèmes internes (les capteurs, le tableau de bord, ...) ainsi que le monde extérieur (les obstacles, les conditions météorologiques, ...). Ces observations sont interprétées et analysées de façon intuitive par le pilote afin de leur donner un sens en utilisant ses propres connaissances sur l'environnement, les objets, le lien entre eux et leurs comportements.

En se basant sur ces observations, et les précédentes aussi, le pilote doit identifier des événements qui peuvent être un élément perturbateur sur la réalisation de la mission. Il doit juger si l'apparition d'un tel événement demande un changement dans le plan de la mission ou pas. Plus précisément, le pilote doit savoir si l'événement doit déclencher une prise de décision ou pas par rapport à la modification du plan de la mission. Donc le pilote ne prend pas des décisions complexes en continu, mais seulement à certains moments qu'il évalue comme critiques.

Une fois que le pilote a décidé de changer le plan de la mission, il est face à un problème de

décision complexe. Il doit réfléchir aux alternatives possibles, à leur impact sur la mission, et à comment les exécuter en pratique. En général, le pilote décompose ce problème en un ensemble de sous-problèmes, en adoptant l'approche « diviser pour mieux régner » [Armstrong *et al.*, 1975]. Il commence par choisir une action de haut niveau parmi un ensemble d'actions possibles (par exemple, atterrir, sauter un waypoint, retourner à la base, ...). Le choix de l'une ou l'autre de ces actions se base sur leur évaluation selon plusieurs critères (risque, coût, objectif de la mission, ...) qui déterminent leur impact sur la mission. Ce choix dépend aussi de la façon opérationnelle dont l'action peut être exécutée (choix de la trajectoire, ...), ce qui constitue en soi une sous-décision.

Une fois que les décisions sont prises, il reste au pilote à exécuter sa décision en effectuant la suite des manœuvres nécessaires. Il est intéressant de noter que dans une même situation, le résultat de ce processus de prise de décision peut varier d'un pilote à l'autre, selon son niveau d'expertise, ses expériences et ses propres jugements de la situation.

Dans la suite, nous proposons un moteur de décisions pour des drones autonomes qui s'inspire du raisonnement adopté par un pilote à bord d'un aéronef durant le vol.

3.2 Moteur de décisions multi-niveaux

En nous basant sur la logique adoptée par le pilote lors d'une mission, nous proposons de décomposer le processus de décision du drone autonome pendant la mission en plusieurs niveaux. Cette décomposition est aussi justifiée par le besoin d'avoir un modèle lisible où une décision complexe est décomposée en un ensemble de sous-décisions moins complexes ce qui facilite la tâche d'expliquer à l'opérateur le fonctionnement du moteur ainsi que les décisions que le drone aura prises. Cela permet à l'opérateur d'avoir plus de confiance dans ce moteur de décisions qui sera intégré dans le drone et donc d'avoir plus de confiance dans le comportement du drone. Une autre raison qui nous guide vers un moteur multi-niveaux, est d'avoir un modèle de décision qui garantit une certaine traçabilité et un haut niveau de compréhension de la décision. C'est-à-dire dans le cas où le pilote veut avoir plus d'explications sur une prise de décision, il doit être possible de tracer l'origine de cette décision et de comprendre pourquoi elle a été prise.

Comme présenté sur la figure 3.1, le moteur de décisions que nous proposons est composé de deux niveaux de prise de décision de haut niveau. Le premier a pour but de décider si l'apparition d'un événement nécessite un changement de plan de la mission ou pas, ce qui concrètement signifie le déclenchement ou non du niveau 2. Ce dernier, une fois déclenché permet de choisir une action de haut niveau à exécuter. En plus de ces deux niveaux, un module de calcul de trajectoire fait aussi partie de ce moteur. Chacun de ces trois éléments, qui sont détaillés dans la suite, dépend d'un contexte [Ajami *et al.*, 2013], qui est un ensemble d'éléments qui décrivent l'environnement du drone.

Le contexte peut inclure des informations sur la mission, son objectif (par exemple, surveiller une cible, protéger un convoi, ...) ou son état actuel. D'autres informations relatives au drone sont également incluses dans ce contexte qui sont fournies par les capteurs du drone. Les sorties des capteurs peuvent être utilisées directement (par exemple coordonnées GPS, altitude) ou peuvent être traitées avant (les petits pignons noirs sur les flèches vertes), tandis que d'autres informations concernant son environnement peuvent également être utilisées (carte des zones de vol, zone d'invisibilité, conditions météorologiques).

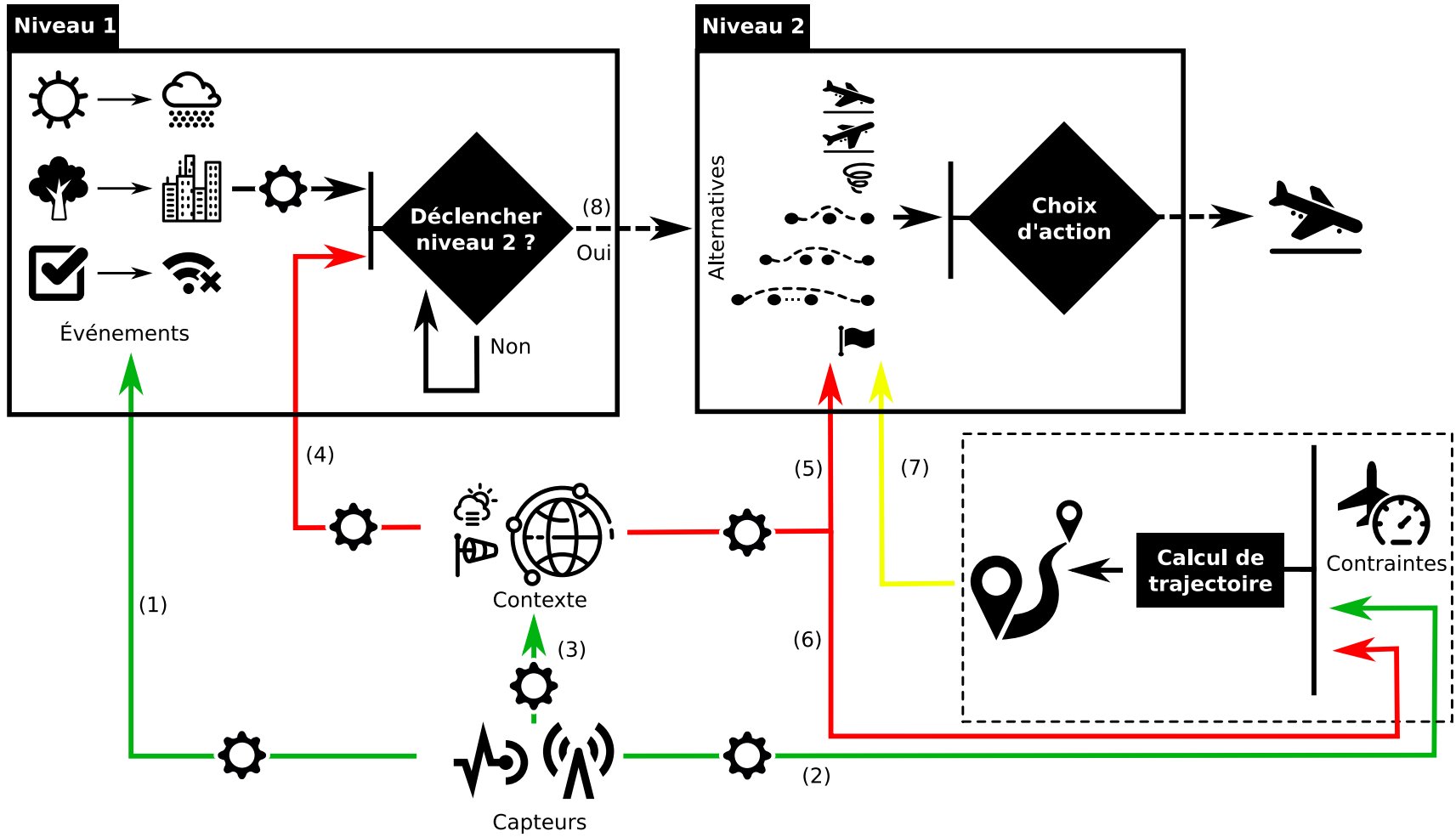


Figure 3.1 – Moteur de décisions multi-niveaux

Ce contexte peut influencer sur le changement de plan de mission puisqu'il intervient dans l'évaluation de l'événement à son apparition. Cela est indiqué par la flèche rouge (4) dans la figure 3.1. Le choix de l'action de haut niveau dans le niveau 2 est aussi influencé par le contexte étant donné qu'il intervient dans l'évaluation de chaque alternative (par exemple, choisir de retourner à la base, n'a pas les mêmes conséquences dans un contexte de météo défavorable, que dans un contexte de temps calme). Ceci est représenté par la flèche rouge (5) qui va aux alternatives dans la figure 3.1.

Le niveau 1 permet de suivre la réalisation de la mission et toutes les informations susceptibles d'avoir un impact sur son succès. En fonction du contexte du drone (flèche rouge (4)) et de l'occurrence de certains événements, le deuxième niveau peut être déclenché (flèche noire pointillée (8)). Ces événements peuvent être liés à l'environnement du drone (par exemple, un changement de zone de vol, l'apparition d'un obstacle, la détection de fortes pluies, une panne mécanique, ...) ou à des niveaux de risque (par exemple, dépasser un certain seuil de risque pour le drone ou pour la mission ou pour l'environnement, ...). Ils sont, en général, alimentés par des informations données par les capteurs (flèche verte (1)).

Le niveau 2 consiste à déterminer quelle action de haut niveau (par exemple *décoller*, *continuer*, *sauter* un ou plusieurs waypoints, *retourner à la base*, *tourner autour* d'un waypoint, *atterrir*, ...) est la meilleure réponse aux risques générés par les événements de la première couche. Classiquement, ce choix se fait selon un moteur de règles expertes, qui évaluent les actions, selon le contexte dans lequel se trouve la mission (flèche rouge (5)), et les trajectoires potentielles que le drone devra suivre (flèche jaune (7)).

Afin d'évaluer les trajectoires possibles pour les actions potentielles, un module de calcul de trajectoire (rectangle en pointillé noir) est également inclus dans notre moteur de décisions. Il permet de générer une trajectoire pour exécuter une action donnée. Des contraintes liées au drone sont prises en compte, par exemple des contraintes physiques (angle de montée, angle de rotation, ...), des contraintes aérodynamiques (portance, traînée, ...) ainsi que les contraintes énergétiques et temporelles.

Ce moteur de décisions répond à une partie de la description du processus de prise de décision d'un pilote humain. L'observation de l'environnement du drone est assurée par le contexte et les capteurs du drone. L'aspect multi-étapes où le pilote ne prend pas de décisions de façon continue, mais uniquement lorsque c'est nécessaire est représenté par le niveau 1. La décomposition des décisions complexes en un ensemble de sous-décisions moins complexes se retrouve dans l'identification des actions possibles au niveau 2. Cela dépend du contexte du drone et l'évaluation de l'impact de leur exécution en utilisant la trajectoire obtenue par le module de calcul de trajectoire.

L'originalité de ce moteur réside dans la possibilité d'intégrer des composants classiques de décision comme les réseaux de neurones, les arbres de décisions (voir chapitre 2), les bases de règles expertes, ... tel qu'on peut les retrouver actuellement dans des drones autonomes.

Une des limites de cette proposition est le fait que les décisions ne sont pas « personnalisées » selon chaque opérateur. Comme déjà mentionnées dans la section 3.1 pour une même situation, différentes décisions peuvent être prises selon le pilote. Nous proposons donc dans la suite une évolution de ce premier moteur de décisions qui prend en considération cet aspect en intégrant les préférences d'un pilote dans la prise de décision. Le but de cette intégration est de donner au pilote plus de confiance aux décisions prises par le drone puisque ce dernier va se comporter selon ses attentes.

3.3 Intégration des préférences d'un pilote

Dans le moteur de décisions présenté dans la section précédente (figure 3.1) la prise de décision intervient à différents endroits : dans le niveau 1, le niveau 2 et même dans le module de calcul de trajectoire. Afin d'augmenter la confiance qu'un opérateur peut avoir dans les décisions prises par un drone autonome, nous proposons d'intégrer, selon les besoins, des modèles de préférences d'un opérateur humain, dans chacun des 3 niveaux proposés.

Nous augmentons donc le schéma qui représente le moteur de décisions (figure 3.2) en y intégrant l'opérateur (en haut, en bleu). Les préférences réelles de cet opérateur sont représentées par un modèle, qui est configuré à partir de techniques d'élicitation des préférences (pignons noirs sur les flèches bleues). Les algorithmes de décision plus classiques (moteur de règles, ...) sont alors remplacés par des algorithmes d'AMCD, dans lesquels de multiples critères de décision peuvent être pris en compte. Nous illustrons cela par les matrices grisées qui représentent les performances des différentes alternatives selon les critères retenus. Afin d'utiliser les algorithmes d'AMCD, les modèles des préférences de l'opérateur ainsi que leurs paramètres préférentiels correspondants sont alors injectés en amont dans le moteur avant la mission. .

Dans la suite nous discutons de l'intégration de techniques d'AMCD dans les 3 couches du moteur de décisions générique que nous proposons.

3.3.1 Intégration des préférences dans le niveau 1

3.3.1.1 Modélisation du problème de décision

Ce niveau sert à surveiller l'état du drone ainsi que son environnement afin de détecter tout événement qui pourrait perturber la réalisation de la mission. L'ensemble de ces événements sont les alternatives pour notre problème de décision :

- un changement de zone de vol,
- l'apparition d'un obstacle,
- un dépassement d'un certain seuil de risque pour le drone,
- un dépassement d'un certain seuil de risque pour l'environnement,
-

Lors de l'apparition d'un tel événement, de multiples critères sont utilisés pour évaluer son impact sur la mission. Ces critères peuvent être différents selon la formulation du problème de décision et les priorités ou l'expertise de l'opérateur. Nous pouvons citer quelques exemples :

- le risque sur le drone associé à chaque événement,
- le risque sur l'environnement associé à chaque événement,
- la réalisation de la mission, qui est un pourcentage pondéré des sous-objectifs atteints,
- une catégorie de l'événement (événement intérieur au drone, extérieur au drone, ...),
-

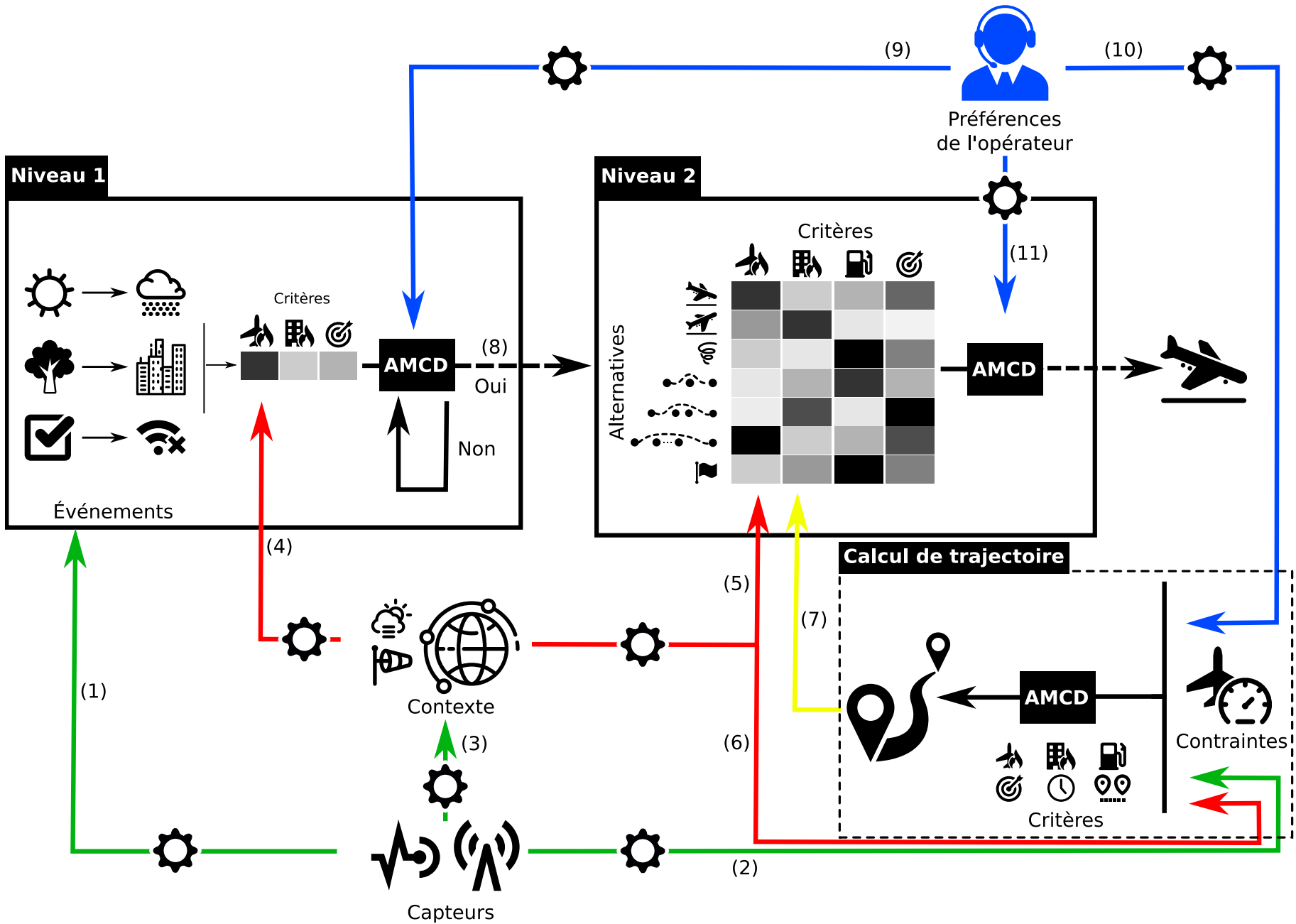


Figure 3.2 – Intégration des préférences de l'opérateur dans le moteur de décisions

Le but ici est d'identifier si un événement (donc une alternative de décision) est critique ou non. Cette évaluation binaire (critique / non critique), selon de multiples critères, nous mène donc à intégrer dans cette couche du moteur un modèle de tri multi-critère.

Comme indiqué dans le chapitre 2, le choix du modèle d'AMCD est guidé par la formulation de notre problème de décision. Les alternatives ici sont les événements qui apparaissent lors de la mission qui sont évalués sur des critères avec des échelles d'évaluation qui peuvent être hétérogènes, qualitatives (le risque par exemple) et quantitatives (l'exécution de la mission par exemple). Cela nous mène à des modèles de tri basés sur le surclassement. Une conséquence de ce choix est que ces modèles permettent assez facilement de générer une règle qui synthétise les raisons expliquant pourquoi une alternative a été affectée à une certaine catégorie. Cela répond donc par ailleurs à notre souci de lisibilité du résultat, et doit contribuer à la confiance accrue de l'opérateur dans les décisions du drone.

Plusieurs modèles de surclassement répondent aux contraintes énoncées ci-avant. On peut par exemple citer ELECTRE TRI [Yu, 1992, Roy et Bouyssou, 1993], PROMETHEE TRI [Figueira *et al.*, 2004] et MR-Sort [Bouyssou et Marchant, 2007a, Bouyssou et Marchant, 2007b]. Ces modèles sont présentés dans le chapitre 2 et nous détaillons l'utilisation de MR-Sort dans le contexte de notre travail ci-dessous.

3.3.1.2 Instanciation dans notre moteur

Dans notre moteur, nous choisissons d'utiliser un modèle MR-Sort avec veto. Une des raisons pour ce choix est l'explicabilité de ce modèle, et le nombre de paramètres préférentiels assez restreints nécessaires pour le calibrer. En effet, l'opérateur du drone a besoin de valider le modèle de décision sachant qu'il n'est pas forcément un expert du domaine d'AMCD. Pour cette raison, notre choix se porte sur un modèle qui est facile à expliquer et dont les recommandations sont faciles à interpréter.

Pour rappel, l'évaluation des événements (donc des alternatives) dépend du contexte et s'effectue dans notre proposition selon les 3 critères suivants (flèche rouge (4) de la figure 3.2) :

- le risque sur le drone associé à chaque événement,
- le risque sur l'environnement associé à chaque événement,
- la réalisation de la mission, qui est un pourcentage pondéré des sous-objectifs atteints.

MR-Sort avec veto est un modèle de surclassement pour les problèmes de tri. Il est basé sur une version simplifiée du modèle ELECTRE TRI [Roy, 1991, Mousseau *et al.*, 2000, Figueira *et al.*, 2005] et proche de la version axiomatisée en [Bouyssou et Marchant, 2007a, Bouyssou et Marchant, 2007b]. Il est défini par l'ensemble des paramètres suivants :

- m le nombre de critères,
- w_j est le poids du critère j , avec $\sum_1^m w_j = 1$
- k est le nombre des catégories,
- $\{c_1, \dots, c_k\}$ les k catégories,
- $\mathcal{L} = \{l_0, \dots, l_k\}$ l'ensemble des limites des catégories,
- $\lambda \in]0.5, 1]$ est un seuil de majorité.
- $v_j^h, j = 1, \dots, m, h = 1, \dots, k$ les seuils de veto pour chaque limite de catégorie h et chaque critère j .

Le problème de décision que nous considérons est aussi défini par les éléments suivants :

- \mathcal{A} un ensemble fini d'alternatives, c'est-à-dire les événements que nous évoquions précédemment,
- $\mathcal{M} = \{1, \dots, m\}$ l'ensemble des indices des critères,
- $a_j, j \in \mathcal{M}$ est l'évaluation de a sur le critère j , $a \equiv (a_1, \dots, a_m) \forall a \in \mathcal{A}$
- $l_{h,j}, j \in \mathcal{M}, h = 0, \dots, k$ est l'évaluation de l_h sur le critère j , $l_h \equiv (l_{h,1}, \dots, l_{h,m})$.

Les alternatives doivent être affectées à une des k catégories $\{c_1, \dots, c_k\}$. Ces catégories sont ordonnées selon les préférences du décideur, c'est-à-dire c_1 est la moins préférée, alors que c_k est la préférée. Chaque catégorie c_h est définie par sa limite inférieure l_{h-1} et sa limite supérieure l_h . l_0 , la limite fictive inférieure, est définie par les pires évaluations de chaque critère, de sorte que toute alternative puisse être affectée au moins à la catégorie c_1 . l_k , la limite fictive supérieure, est définie par les meilleures évaluations de chaque critère, donc toute alternative sera affectée au mieux à la catégorie c_k . Pour la présentation qui suit nous supposons, sans perte de généralité que le décideur préfère les valeurs élevées sur les critères. En outre, $\forall j \in \mathcal{M}, h = 1, \dots, k, l_{h-1,j} \leq l_{h,j}$.

Deux règles pour attribuer une alternative à une classe peuvent être trouvées dans la littérature, les règles d'attribution pessimiste et optimiste, parmi lesquelles la première est la plus couramment utilisée. Dans ce cas, une alternative a est attribuée à la catégorie c_h la plus élevée possible, de sorte que a surclasse la limite inférieure de la catégorie l_{h-1} et aucun critère ne s'oppose fermement (oppose son veto v_j^{h-1}) à cette affirmation.

Dans un modèle MR-Sort, une alternative a surclasse la limite inférieure de la catégorie l_{h-1} ($a \text{ S } l_{h-1}$) s'il existe une majorité de critères sur lesquels a est au moins aussi bon que l_{h-1} . Plus précisément, si :

$$\sum_{j \in \mathcal{M}} C(a, l_{h-1}) w_j \geq \lambda \text{ et } a_j \geq v_j^{h-1} \forall j \in \mathcal{M}$$

avec λ est le seuil majoritaire et $C(a, l_{h-1}) \in [0, 1]$ est le degré de concordance global calculé à partir des degrés de concordances locaux $C_j(a, l_{h-1})$:

$$C(a, l_{h-1}) = \sum_{j \in \mathcal{M}} w_j C_j(a, l_{h-1}), \text{ où}$$

$$C_j(a, l_{h-1}) = \begin{cases} 1, & \text{si } a_j \geq l_{h-1,j} \\ 0, & \text{sinon.} \end{cases} \quad \forall a \in \mathcal{A}, j \in \mathcal{M}, h = 1, \dots, k+1,$$

Si $\sum_{j \in \mathcal{M}} C_j(a, l_{h-1}) w_j < \lambda$ ou $\exists j$ tel que $a_j < v_j^{h-1} \forall j \in \mathcal{M}$ alors a ne surclasse pas l_{h-1} ($a \not\text{S } l_{h-1}$).

En résumé, la règle d'affectation pour MR-Sort est la suivante :

$$a \in c_h \iff a \text{ S } l_{h-1} \wedge a \not\text{S } l_h$$

Comme nous l'avons déjà mentionné, les paramètres d'un modèle MR-Sort sont fixés à partir des préférences exprimées par l'opérateur. En effet ces paramètres sont inférés dans une phase d'élicitation des préférences. Plusieurs travaux dans la littérature ont été déjà proposés pour inférer les paramètres des modèles de tri basés sur le surclassement qui ont déjà été présentés dans la section 2.4.1.

3.3.1.3 Exemple illustratif

Pour bien comprendre la procédure d'affectation d'une alternative à une catégorie avec un modèle MR-Sort, nous présentons un exemple illustratif avec notre cas d'application de drone. Nous supposons ici que les paramètres du modèle MR-Sort sont déjà fixés dans la phase d'élicitation des préférences.

Nous considérons que le drone est en mission et en arrivant à un certain moment le système détecte un événement perturbateur. Cet événement est évalué selon les trois critères énoncés précédemment : deux critères de risque pour le drone R_{UAV} et pour l'environnement R_{Env} (selon une échelle d'évaluation qualitative faible \prec moyen \prec haut) et le critère de déroulement de la mission M sur une échelle quantitative qui exprime un pourcentage d'atteinte de tous les objectifs (à maximiser).

Le but est d'affecter l'événement perturbateur à une des 2 catégories « critique » (C_1) ou « non critique » (C_2). La catégorie « critique » déclenchera la prise de décision du niveau 2, alors que la catégorie « non critique » non, et la mission se poursuivra normalement.

Les paramètres du modèle MR-Sort sont présentés dans le tableau 3.1.

	R_{UAV}	R_{Env}	M
l_1	moyen	moyen	30%
w	0.3	0.4	0.3
$veto$	-	-	25%
λ		0.7	

Tableau 3.1 – Les paramètres du modèle MR-Sort

Dans notre exemple nous avons un seul veto sur le critère M : $v_M = 25\%$. En effet une alternative qui ferait chuter l'accomplissement de la mission à moins de 25% est automatiquement affectée à la classe « critique », sans tenir compte des valeurs sur les autres critères.

À un moment t_1 de la mission, un événement x est détecté. Il est évalué comme suit : $x \equiv (faible, moyen, 27\%)$. Pour évaluer la catégorie à laquelle l'événement x est affecté, nous calculons d'abord son degré de concordance par rapport au profil l_1 :

$$C(x, l_1) = 1 \cdot 0.3 + 1 \cdot 0.4 + 0 \cdot 0.3 = 0.7$$

Nous constatons que $C(x, l_1) \geq \lambda$ et $x_M = 27\% \geq 25\%$. Donc l'événement $x \in c_2$ et il ne déclenche pas le niveau 2.

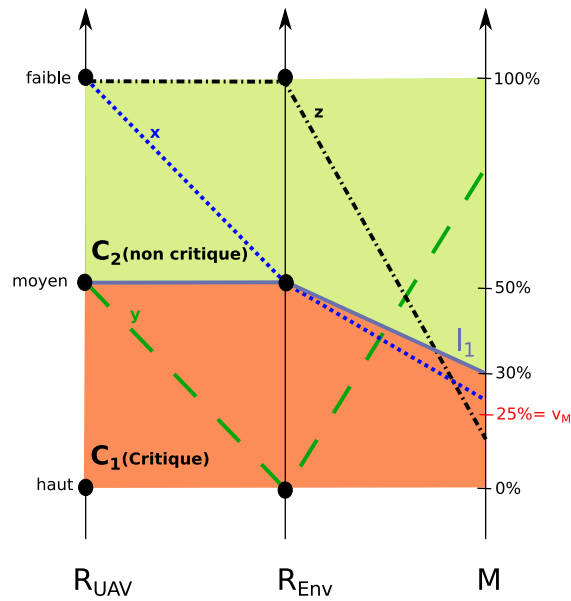


Figure 3.3 – Présentation des événements dans le modèle MR-Sort

À un moment t_2 de la mission, un autre événement y est observé. Il est évalué par $y \equiv (\text{moyen}, \text{haut}, 80\%)$. Nous commençons par calculer son degré de concordance par rapport au profil l_1 :

$$C(y, l_1) = 1 \cdot 0.3 + 0 \cdot 0.4 + 1 \cdot 0.3 = 0.6$$

$C(y, l_1) < \lambda$ et $y_M = 80\% \geq 25\%$, donc $y \in c_1$, et y est considéré comme critique parce que son degré de concordance est inférieur au seuil. Donc l'événement y déclenche le niveau 2.

À un moment t_3 de la mission un autre événement z apparaît. Il est évalué comme suit $z \equiv (\text{faible}, \text{faible}, 20\%)$.

Nous avons le degré de concordance par rapport au profil l_1 :

$$C(z, l_1) = 1 \cdot 0.3 + 1 \cdot 0.4 + 0 \cdot 0.3 = 0.7$$

Nous avons, $C(z, l_1) \geq \lambda$ mais $z_M = 20\% < 25\%$. Donc $z \in c_1$ et z est considéré comme critique à cause de l'accomplissement de la mission qui chuterait à moins de 25%. Cela permet de déclencher le niveau 2.

Selon ces paramètres du modèle MR-Sort, si les événements y et z apparaissent lors de la mission ils sont considérés comme critiques par le système ce qui déclenche la prise de décision du niveau 2. Par contre si l'événement x apparaît il est considéré comme événement non critique et donc il n'y a pas de changement au niveau du plan de mission actuel.

3.3.2 Intégration des préférences dans le niveau 2

3.3.2.1 Modélisation du problème de décision

Une fois que la prise de décision de ce niveau est déclenchée par un événement détecté par le niveau 1, le drone doit choisir une action parmi un ensemble d'actions possibles. L'intégration de techniques d'AMCD permet d'évaluer ces actions sur plusieurs critères, tout en intégrant les préférences de l'opérateur.

Quelques exemples d'alternatives de haut niveau pour ce second niveau sont :

- continuer,
- retourner à la base,
- sauter ou plusieurs waypoints,
- atterrir d'urgence,
- survoler un waypoint,
-

Ces actions dépendent du contexte dans lequel se trouve le drone. Elles peuvent donc différer d'un contexte à un autre, parce qu'elles ne sont pas potentiellement réalisables par le drone. Par exemple l'action atterrir n'est pas envisageable lorsque le drone survole la mer ou sauter deux waypoints ne fait pas partie des actions possibles s'il n'y a pas assez de carburant pour l'exécuter.

Ces actions sont naturellement évaluées selon de multiples critères, en fonction des conséquences qu'elles ont sur la mission, le drone ou l'environnement. Ces critères peuvent varier selon le type de mission, l'opérateur, ou le contexte général de la mission. Une liste non-exhaustive de critères est donnée ci-après :

- le risque pour le drone [Ruz *et al.*, 2006], c'est-à-dire le risque associé au survol de différentes zones telles que forêts, mer, zones militaires,
- le risque pour l'environnement, tel que les personnes, les bâtiments, en cas d'accident, par exemple,
- la consommation d'énergie [Franco et Buttazzo, 2015], correspondant à la quantité d'énergie restante après l'exécution de l'action sélectionnée,
- la réalisation de la mission, qui est un pourcentage pondéré des sous-objectifs atteints,
- le coût,
- le temps d'exécution,
-

Le contexte et l'état de la réalisation de la mission jouent également un rôle dans les évaluations des alternatives sur l'ensemble des critères. Cela signifie qu'une alternative n'a pas nécessairement la même évaluation d'un contexte à un autre, ou même d'un waypoint à un autre. Par exemple l'évaluation de l'action « continuer » sur le critère consommation d'énergie n'est pas la même dans un contexte avec une mauvaise condition météorologique que dans un contexte où le

drone a perdu l'un de ses capteurs.

L'importance de ces critères, c'est-à-dire des conséquences des actions de décision possibles sur la mission, peut être différente d'un pilote à un autre. Par conséquent, un modèle des préférences du pilote doit être choisi, intégré dans le choix de ces alternatives, et calibré selon les préférences de l'opérateur en amont de la mission.

Le choix du modèle d'AMCD est ici guidé dans un premier temps par les caractéristiques des critères choisis. Les critères, parmi ceux de la liste d'exemple précédente, peuvent avoir des échelles d'évaluation hétérogènes, qualitatives comme le risque ou quantitatives, comme le coût par exemple. Cela nous guide vers des modèles qui gèrent ce problème d'hétérogénéité et comme nous avons vu dans le chapitre 2 les méthodes MAVT nécessitent une étape supplémentaire pour rendre les échelles commensurables. Nous préférons ici choisir un modèle de surclassement, étant donné que cette classe de modèles gère les échelles hétérogènes sans cette étape supplémentaire.

Un grand nombre de modèles de surclassement ont été proposés pour traiter les différentes problématiques de décision : tri, choix et classement. Dans notre cas, où le drone doit exécuter une action, la problématique du tri n'est pas adaptée. En effet, celle-ci permet de partitionner l'ensemble des alternatives possibles en plusieurs catégories, et potentiellement plusieurs alternatives pourraient se retrouver dans la catégorie des alternatives préférées. Cela impliquerait donc que le drone devrait choisir au hasard parmi ces alternatives, ce qui ne contribuera pas à augmenter la confiance que l'opérateur aurait dans les décisions du drone. Pour la problématique de choix, où une seule action est choisie, le drone peut avoir des imprévus lors de l'exécution de l'action choisie. Pour cette raison il a besoin d'avoir plus qu'une seule action en cas de problème d'exécution. Cela nous mène à traiter une problématique de classement où les actions possibles sont ordonnées selon un (pré)-ordre total (possiblement partiel). Dans une telle situation, le drone retiendra l'action préférée, et si elle s'avère non réalisable, il prendra la suivante dans le classement. En cas d'ex-aequo (donc d'indifférence) dans ce classement, le drone choisira au hasard.

Plusieurs modèles de surclassement pour les problèmes de classement sont proposés dans la littérature. On peut citer les méthodes ELECTRE II-IV, PROMETHEE I-IV et RMP qui sont donc adaptées pour ce niveau de décision dans le moteur et que nous avons introduit dans le chapitre 2 d'état de l'art.

3.3.2.2 Instanciation dans notre moteur

Pour notre moteur de décisions, nous avons choisi le modèle SRMP (Simple Ranking with Multiple Reference Profiles) parce que dans le contexte des drones autonomes, le pilote a besoin d'avoir confiance dans les décisions prises par le drone et donc de bien comprendre le modèle de décision. Les méthodes ELECTRE et PROMETHEE souffrent de la dépendance à la tierce alternative. Cela rend leurs classements discutables, alors que pour SRMP il y a indépendance vis-à-vis des tierces alternatives. Par ailleurs, le résultat de SRMP peut être expliqué sans trop de difficultés à un non-expert du domaine (c'est-à-dire l'opérateur).

Pour rappel, dans les méthodes de surclassement, une relation « au moins aussi bonne que » se construit entre des paires d'alternatives évaluées sur des critères multiples. Cette relation binaire, appelée « relation de surclassement » [Roy, 1991] est souvent désignée par \succsim .

Une alternative a surclasse une autre, b , autrement dit $a \succsim b$, s'il existe des arguments suffisamment solides pour déclarer que a est au moins aussi bonne que b et s'il n'y a aucune raison essentielle pour réfuter cette affirmation.

Cependant, la comparaison de toutes les alternatives possibles en fonction d'une telle relation peut entraîner des cycles dans la relation de surclassement, ce qui rend impossible la création d'un ordre complet. Il a donc été proposé par Rolland [Rolland, 2013b] d'utiliser un profil de référence dans la comparaison de deux alternatives : a est considéré comme strictement préféré à b si et seulement si la relation de surclassement entre a et le profil de référence est plus forte que la relation de surclassement entre b et profil de référence.

Formellement, nous désignons à nouveau par \mathcal{A} un ensemble de n alternatives et par $M = \{1, \dots, m\}$ les indices des m critères. L'évaluation d'une alternative $a \in \mathcal{A}$ sur le critère $j \in M$ est notée a_j et prend sa valeur dans l'échelle d'évaluation associée X_j , $a \equiv (a_1, \dots, a_m) \in \mathcal{X}$ où \mathcal{X} est l'espace d'évaluation des critères $\mathcal{X} = \prod_{j \in M} X_j$. À chaque critère, il est associé un préordre \succsim_j , tel que si le décideur considère que a est au moins aussi bon que b sur j alors $a_j \succsim_j b_j$.

Le modèle SRMP est défini par plusieurs paramètres, dont les valeurs peuvent différer d'un décideur à l'autre et qui doivent être identifiés au préalable, soit directement, soit par un algorithme d'apprentissage. Ces paramètres sont :

- les k profils de référence : $\mathcal{P} = \{p^h, h = 1..k\}$ qui se dominent mutuellement, soit si $p^h \equiv (p_1^h, \dots, p_j^h, \dots, p_m^h)$ correspond aux évaluations de p^h sur les m critères, alors :

$$p_j^h \succsim_j p_j^l, \forall h, l \in 1..k, h > l$$

- l'ordre lexicographique des profils : σ , qui correspond à une permutation sur $1..k$ et qui représente l'ordre des profils qui seront utilisés pour la comparaison des alternatives ;
- les poids des critères : w_1, w_2, \dots, w_m , où $w_j \geq 0$ et $\sum_{j \in M} w_j = 1$.

Lorsque nous comparons deux alternatives $a, b \in \mathcal{A}$ par rapport à un profil p^h , $h \in \{1..k\}$, deux situations peuvent se présenter :

- a est strictement préféré à b par rapport à p^h , c'est-à-dire

$$a \succ_{p^h} b \Leftrightarrow \sum_{j \in C(a, p^h)} w_j > \sum_{j \in C(b, p^h)} w_j ;$$

- a est indifférent à b par rapport à p^h , c'est-à-dire

$$a \sim_{p^h} b \Leftrightarrow \sum_{j \in C(a, p^h)} w_j = \sum_{j \in C(b, p^h)} w_j$$

où $C(a, p^h) = \{j \in M : a_j \succsim_j p_j^h\}$ est l'ensemble des critères sur lesquels $a \in \mathcal{A}$ est au moins aussi bon que p^h .

Pour classer $a, b \in \mathcal{A}$ en utilisant la procédure du modèle SRMP, nous considérons séquentiellement les profils $p^{\sigma(1)}, p^{\sigma(2)}, \dots, p^{\sigma(k)}$ selon l'ordre lexicographique comme ceci :

- a est strictement préféré à b ssi :

$$\begin{aligned} & a \succ_{p^{\sigma(1)}} b \text{ ou} \\ & a \sim_{p^{\sigma(1)}} b \text{ et } a \succ_{p^{\sigma(2)}} b \text{ ou} \\ & \dots \\ & a \sim_{p^{\sigma(1)}} b \text{ et } \dots \text{ et } a \sim_{p^{\sigma(k-1)}} b \text{ et } a \succ_{p^{\sigma(k)}} b \end{aligned}$$

- a est indifférent à b ssi :

$$a \sim_{p^{\sigma(1)}} b \text{ et } \dots \text{ et } a \sim_{p^{\sigma(k)}} b$$

Intuitivement, a est préféré à b dès que l'on rencontre un profil dans l'ordre lexicographique pour lequel a est préféré à b . Sinon, a et b sont considérés comme indifférents, ce qui signifie qu'aucun profil n'a pu discriminer entre a et b .

Pour définir les paramètres d'un modèle SRMP, il est nécessaire d'interagir avec l'opérateur, afin d'éliciter ses préférences. Nous détaillons dans le chapitre 5 et le chapitre 6 les méthodes utilisées pour éliciter les paramètres d'un modèle SRMP.

3.3.2.3 Exemple illustratif

Nous considérons ici que le niveau 1 a évalué un événement comme critique, et qu'ensuite le niveau 2 est déclenché. Trois actions sont possibles, x, y et z, qui sont évaluées selon trois critères : risque (R) (faible \succ_R moyen \sim_R haut \prec_R très haut), énergie restante (E) (en %, un niveau élevé est préférable à un niveau plus bas) et progrès de mission (M) (en %, plus c'est haut, mieux c'est). Les évaluations des alternatives sur ces trois critères sont présentées dans le tableau 3.2, ainsi que les paramètres de préférence, qui ont déjà été élicités en amont de la mission, et intégrés dans le moteur de décisions du drone.

	R	E	M
x	faible	80%	20%
y	haut	55%	90%
z	moyen	20%	50%
p^1	haut	35%	40%
p^2	faible	70%	80%
w	1/3	1/3	1/3
σ	(1, 2)		

Tableau 3.2 – Évaluations des alternatives et les paramètres du modèle SRMP

Les deux profils de référence permettent de définir trois segments sur les performances de chaque critère : meilleur que p^2 ; entre p^1 et p^2 ; pire que p^1 . En d'autres termes, les profils

de référence permettent d'identifier un encodage ordonné pour chaque critère défini par trois intervalles de performances ordonnés (A , B et C) comme l'illustre la Fig. 3.4, de telle manière que :

- les performances supérieures à p^2 pour chaque critère sont notées A (ce qui peut être interprété comme « bon »).
- les performances entre p^1 et p^2 sur chaque critère sont notées B (ce qui peut être interprété comme « moyen »).
- les performances inférieures à p^1 pour chaque critère sont notées C (interprétées comme « mauvais »).

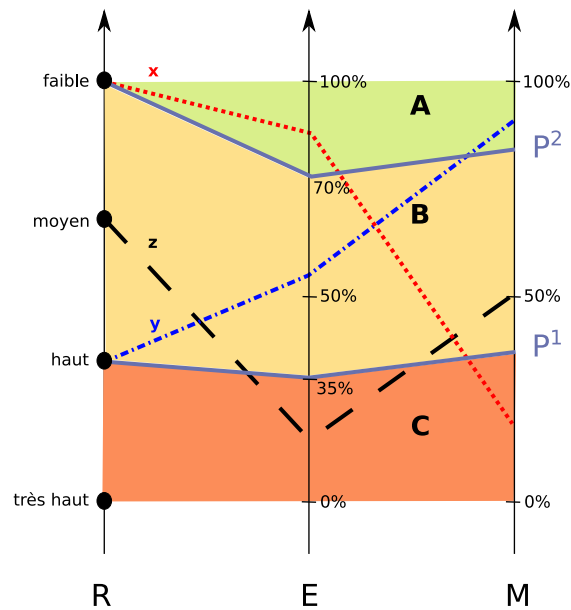


Figure 3.4 – Exemple de modèle SRMP

Les évaluations des alternatives par cet encodage sont présentées dans le tableau 3.3.

	R	E	M
x	A	A	C
y	B	B	A
z	B	C	B

Tableau 3.3 – Encodage des évaluations des alternatives

Pour classer les trois alternatives selon ce modèle SRMP, nous comparons d'abord chacune des alternatives aux autres en utilisant les profils dans l'ordre lexicographique $\sigma = (1, 2)$.

Pour p^1 :

$$\left. \begin{array}{l} \sum_{j \in C(x, p^1)} w_j = 1/3 + 1/3 + 0 = 2/3 \\ \sum_{j \in C(y, p^1)} w_j = 1/3 + 1/3 + 1/3 = 1 \\ \sum_{j \in C(z, p^1)} w_j = 1/3 + 0 + 1/3 = 2/3 \end{array} \right\} \Rightarrow \begin{array}{l} y \succ_{p^1} x \\ y \succ_{p^1} z \\ x \sim_{p^1} z \end{array}$$

Le profil p^1 n'est pas capable de discriminer entre x et z , et donc nous continuons avec le profil p^2 , en respectant l'ordre lexicographique.

Pour p^2 :

$$\left. \begin{array}{l} \sum_{j \in C(x, p^2)} w_j = 1/3 + 1/3 + 0 = 2/3 \\ \sum_{j \in C(z, p^2)} w_j = 0 + 0 + 0 = 0 \end{array} \right\} \Rightarrow x \succ_{p^2} z$$

Le classement final est ainsi $y \succ x \succ z$, donc y est considéré globalement comme la meilleure alternative, suivie de x et ensuite z . En d'autres termes, le drone choisira l'action y . Si un facteur externe l'en empêche (contrainte physique, ...), il aura la possibilité de choisir x , et ainsi de suite.

3.3.3 Intégration des préférences dans le calcul de trajectoire

Les alternatives de décision du second niveau dépendent fortement de la trajectoire qui sera mise en œuvre pour l'exécuter. Dans cette section nous discutons de l'intégration des préférences de l'opérateur dans le calcul de ces trajectoires, qui permettent ensuite de contribuer à déterminer :

- l'ensemble des alternatives réalisables, étant donné le contexte de la décision de niveau 2,
- les évaluations des alternatives sur certains critères du niveau 2 (comme par exemple la consommation).

La prise de décision au cours de la planification de trajectoire autonome exige la sélection de la trajectoire sans collision la plus appropriée possible par rapport à $m \geq 1$ critères. [Gigerenzer et Todd, 1999] montrent que les pilotes prennent en compte de multiples critères dans les processus décisionnels de la vie réelle. En plus, chaque pilote peut avoir sa propre perception de l'environnement et de l'importance relative de chaque critère et donc de chaque trajectoire possible.

Par conséquent, [Narayan *et al.*, 2013] proposent l'utilisation des techniques d'AMCD au cours du calcul de trajectoire qui permettent de choisir une trajectoire qui reflète mieux les priorités de l'opérateur. Pour ce problème de décision, les alternatives sont l'ensemble des trajectoires possibles pour aller d'un waypoint W_i à un waypoint W_{i+1} . Ces trajectoires sont sous la forme d'une succession des vecteurs en 3 dimensions au cours du temps.

Ces alternatives sont évaluées sur plusieurs critères qui caractérisent leur aspect physique :

- la distance euclidienne entre W_i et W_{i+1} ,
- l'angle vers W_{i+1} ,
- le risque de collision,

- l'altitude entre la position courante et la cible (W_{i+1}),
- la précision de la trajectoire pour atteindre W_{i+1} .

Afin de déterminer quelle alternative est la plus appropriée pour l'opérateur, un classement de toutes les alternatives est effectué. Cela permet à l'opérateur d'accéder directement non seulement à la « meilleure » trajectoire, mais aussi aux trajectoires restantes et à leur classement.

L'approche MAVT est utilisée ici avec une fonction de valeur additive pondérée. Pour chaque trajectoire possible, t l'objectif est de calculer une valeur $V(t)$ tel que :

$$V(t) = \sum_{j=1}^m w_j v_j(t)$$

où w_j est le poids du critère j et $v_j(t)$ est la fonction de valeur marginale qui détermine la valeur attribuée à t sur le critère j .

Pour calibrer ce modèle, les préférences du décideur sont élicitées à l'aide de la méthode UTA [Jacquet-Lagreze et Siskos, 1982, Siskos *et al.*, 2005]. Elle détermine l'ensemble des fonctions de valeur ainsi que les poids des critères à partir de préordres de trajectoires. Ces préordres sont donnés par l'opérateur. Typiquement, il/elle devra ordonner selon ses préférences des trajectoires « fictives » dans un contexte donné, et UTA proposera alors un calibrage des paramètres du modèle additif.

Une nouvelle fois, l'intérêt d'une telle approche est de pouvoir intégrer des critères complémentaires, ou d'ajuster la liste, en fonction des contraintes physiques du drone, des exigences de la mission, ou des préférences de l'opérateur. Sur le schéma de notre proposition (figure 3.2), nous avons par exemple indiqué 5 critères ($m=5$) :

- le risque pour le drone,
- le risque pour l'environnement,
- la consommation d'énergie,
- la distance nécessaire pour arriver à destination,
- le temps nécessaire pour arriver à destination.

3.4 Résumé et mise en perspective

Dans ce chapitre, nous avons décrit une modélisation de la prise de décision d'un pilote humain lors d'une mission en nous basant sur le cycle de Boyd, OODA. Cette modélisation montre que le pilote observe en permanence l'état de l'aéronef (les capteurs, le tableau de bord, ...) et son environnement (les obstacles, la condition météorologique, ...).

Ces observations sont analysées et utilisées pour identifier les événements perturbateurs. C'est-à-dire les événements qui peuvent exiger un changement de plan de la mission. Si un tel événement est détecté, le pilote doit évaluer s'il est nécessaire de faire un changement dans le plan de la mission ou non. Si c'est le cas, cette décision est complexe puisqu'il faut réfléchir à toutes les alternatives possibles ainsi que leur impact sur la réalisation de la mission. Le pilote

décompose ce problème de décision en un ensemble de sous problèmes afin de faciliter la tâche. Dans un premier temps, le pilote doit choisir une action de haut niveau parmi l'ensemble d'actions possibles. Ce choix est fait selon l'évaluation de chaque action sur plusieurs critères (risque, coût, ...)). La façon dont chaque action peut être exécutée est également prise en compte dans cette évaluation, et correspond à une sous-décision.

Nous notons également que cette décision est basée sur le profil du pilote, son expertise et ses préférences. Potentiellement, dans une même situation, deux pilotes différents peuvent donc mettre en œuvre deux décisions différentes.

Un moteur de décisions multi-niveaux est proposé, il est basé sur la description de la prise de décision du pilote lors d'une mission. Tout naturellement nous proposons donc deux niveaux de décision. Le premier niveau permet d'évaluer des événements qui surviennent pendant la mission, et de les classer comme critiques ou non critiques. Le second niveau, s'il est déclenché, sert à évaluer des actions de haut niveau qui permettent de rectifier les conséquences de l'événement qui est apparu. Ce niveau utilise notamment un module de calcul de trajectoire pour estimer certaines de ces conséquences.

Les décisions peuvent dépendre du profil de l'opérateur. Le modèle du moteur de décisions que nous proposons permet ainsi d'être personnalisé selon son profil. À cette fin, nous montrons comment il est possible d'intégrer des techniques d'AMCD dans les différents niveaux du moteur, en prenant en compte les préférences de l'opérateur. Ceci assure que le drone se comporte selon les préférences et priorités de l'opérateur.

Nous argumentons dans ce chapitre le choix des modèles d'AMCD à intégrer dans les différents niveaux, en utilisant les différentes étapes du processus d'aide à la décision. Ces choix sont bien évidemment discutables, mais nous pensons qu'ils répondent au cadre industriel de cette thèse, et sont une réponse adéquate à la nécessité d'explicabilité des décisions prises par le drone.

L'utilisation de ces modèles de préférences nécessite une phase d'élicitation des préférences du pilote en amont de la mission. Cette phase permet de fixer les paramètres des modèles de sorte qu'ils représentent fidèlement les préférences du pilote. Comme nous l'avons montré dans ce chapitre, et le chapitre d'état de l'art, la littérature comporte de nombreux travaux sur l'élicitation des préférences de modèles MR-Sort et de modèles MAVT à fonctions de valeurs additives. De ce fait, nous nous intéressons dans la troisième partie de ce travail à l'élicitation des préférences des modèles SRMP que nous proposons d'utiliser dans le niveau 2.

En plus de proposer un moteur de décisions qui s'adapte aux préférences de l'opérateur, un autre objectif de cette thèse est de réduire l'effort cognitif de l'opérateur pendant la phase d'élicitation des préférences. Dans la partie 3 de ce manuscrit, nous proposons donc également d'étudier un processus d'élicitation incrémentale, dans lequel il s'agit d'utiliser une heuristique de sélection des paires d'alternatives à présenter au pilote à chaque itération du processus. Notre travail consistera à étudier un ensemble de telles heuristiques en vue de déterminer celle qui permet d'atteindre un modèle considéré comme satisfaisant, en confrontant l'opérateur à un nombre minimal de paires d'alternatives dans la phase d'élicitation des préférences.

Chapitre 4

Simulateur de drone autonome intégrant les préférences d'un opérateur

Sommaire

4.1	Motivations et choix	54
4.2	Modélisation du drone	55
4.3	Modélisation d'une mission	59
4.4	Interface utilisateur	63
4.5	Exemple d'exécution de mission	65
4.5.1	Exemple de mission	65
4.5.2	Exécution de la mission dans le simulateur	69
4.6	Résumé et mise en perspective	77

Après avoir proposé un moteur de décisions multi-niveaux qui intègre les préférences d'un opérateur de drone, nous proposons d'en illustrer l'intérêt dans un simulateur, avant une possible intégration dans un vrai drone. Le but de ce chapitre est donc de décrire un tel simulateur de drone autonome où le drone vole selon les préférences d'un opérateur en utilisant le moteur de décisions multi-niveaux. Nous commençons ce chapitre par présenter les motivations du développement de ce simulateur et expliquer le choix d'utiliser MATLAB ainsi que le choix d'utiliser le drone « Thales Watchkeeper WK450 » comme modèle de drone dans le simulateur. Nous présentons ensuite la modélisation de ce drone dans notre simulateur sous la forme d'un point qui est soumis aux mêmes contraintes physiques que le WK450. Ce choix est guidé par le fait que tout d'abord la modélisation exacte du WK450 est très complexe et sort du cadre de notre travail et ensuite aussi par le manque d'informations sur toutes les spécificités de ce drone. La mission est modélisée dans le simulateur par un ensemble de waypoints et d'objectifs. Nous nous concentrons ici sur le niveau 2 plus particulièrement, et nous détaillons donc le calcul de l'évaluation des actions de haut niveau sur les différents critères retenus. Nous décrivons ensuite l'IHM (interface homme-machine) du simulateur, avant de présenter un exemple de mission. Le simulateur est utilisé pour exécuter cet exemple de mission selon les préférences de deux opérateurs différents pour illustrer l'intérêt de notre approche. Cette simulation montre bien que l'intégration du moteur de décisions multi-niveaux avec les préférences d'un opérateur agit sur le comportement du

drone lors de l'exécution de la mission selon ses préférences.

Le simulateur présenté dans ce chapitre s'inspire de notre article [Khannoussi *et al.*, 2019] et a également fait l'objet d'une présentation à la 88ème réunion du Groupe de travail EURO sur l'Aide Multi-Critère à la Décision [Khannoussi *et al.*, 2018c].

4.1 Motivations et choix

Nous avons proposé dans le chapitre 3 un moteur de décisions multi-niveaux qui est basé sur la modélisation de la prise de décision d'un pilote lors d'une mission. Le but est de s'assurer que le drone se comporte selon les priorités et les préférences de l'opérateur pour que ce dernier puisse avoir un niveau de confiance élevé dans les décisions du drone, et être capable de tracer l'origine d'une décision erronée facilement. Dans une situation donnée, ce moteur est donc capable de faire prendre au drone des décisions différentes selon le profil de l'opérateur.

Afin de valider notre proposition avant une éventuelle intégration dans un vrai drone, nous développons un environnement de simulation. Ce simulateur agit comme preuve de concept pour notre modèle de décisions multi-niveaux et montre l'intérêt de notre proposition.

Notre objectif est de développer un simulateur qui permet d'exécuter une mission prédéfinie d'un drone autonome. La mission est définie par un certain nombre de waypoints que le drone doit atteindre, et dans lesquels il doit effectuer une tâche (photo, mesure, ...).

Nous choisissons dans le cadre de ce travail de nous focaliser sur le deuxième niveau du moteur, c'est-à-dire celui où une décision de haut niveau est déclenchée et une action de type atterrir, rentrer à la base, sauter un waypoint, ... doit être choisie. L'arrivée dans un waypoint est donc considérée comme un événement du niveau 1 déclenchant le niveau 2.

Nous choisissons d'implémenter le simulateur en MATLAB [MATLAB, 2018]. Ce dernier donne accès à une variété de différentes boîtes à outils (toolbox). Elles offrent des fonctionnalités qui permettent de faciliter notre tâche de développement. Parmi ces boîtes à outils, nous pouvons citer la Mapping Toolbox car elle permet de manipuler, analyser et visualiser des coordonnées géographiques, dans notre cas les waypoints. Elle inclut des algorithmes pour effectuer des transformations de coordonnées, des calculs basés sur la longitude, la latitude et l'altitude dans divers systèmes de coordonnées et, en option, l'affichage des données dans des vues ou des projections cartographiques personnalisables qui peuvent être en 2 ou 3 dimensions. L'Aerospace Toolbox fournit des outils et des fonctions permettant d'analyser la navigation et l'environnement des véhicules aérospatiaux, ainsi que de visualiser leur vol. Elle permet aussi de représenter l'aérodynamique d'un véhicule et d'intégrer des modèles d'environnement pour l'atmosphère, la gravitation et le vent.

Dans le simulateur, le « Thales Watchkeeper WK450 » (figure 4.1) est utilisé comme modèle de drone. Le « Thales Watchkeeper WK450 » est un drone développé par [Thales, 2010]. Il s'agit d'un drone autonome à longue portée qui a jusqu'à 17h d'autonomie. Potentiellement il peut donc être amené à prendre de nombreuses décisions, et d'effectuer différents types de missions. Donc ce drone a les caractéristiques nécessaires pour illustrer l'intérêt de notre moteur de décisions multi-niveaux.



Figure 4.1 – Thales Watchkeeper WK450 [Thales, 2010]

Développer un modèle fidèle de ce drone est une tâche très complexe et sort du cadre de cette thèse. Nous choisissons donc de représenter ce drone par un point qui est soumis à des contraintes physiques similaires à celles d'un véritable WK450.

Même avec ce choix, des informations très spécifiques liées aux caractéristiques du drone sont nécessaires. Contrairement aux drones commerciaux, les différents paramètres et spécifications des drones militaires ne sont pas toujours disponibles ou ils sont incomplets. Cela nous mène à collecter ces spécifications de différentes sources et à les utiliser pour estimer la valeur des différents paramètres.

Le tableau 4.1 liste les spécifications dont nous disposons pour modéliser le « Thales Watchkeeper WK450 » .

Description	Valeur
Envergure des ailes	10.5m
Longueur	6.1 m
Charge utile	150 kg
Masse au décollage	450 kg
Altitude	5486 m (17999 ft)
Vitesse maximale	176 km/h (95 kts)
Taux de montée	274 m / min (900 ft/min)
Moteur	UEL R802/ 902(W) Wankel rotary 52 HP

Tableau 4.1 – Spécifications du WK450

4.2 Modélisation du drone

Lorsqu'un drone vole, quatre forces principales s'exercent sur son centre de gravité et donc sur le point qui le représente dans notre simulateur : la poussée (T), la traînée (D), le poids (W)

et la portance (L) . Ces quatre forces peuvent être représentées par des vecteurs.

- La poussée (T) est une force qui déplace le drone dans la direction du mouvement. Elle est créée avec l'air aspiré par le moteur puis expulsé dans la direction opposée.
 - La traînée (D) est la force qui agit à l'opposé de la direction du mouvement. Elle a tendance à ralentir le drone. La traînée est causée par le frottement et les différences de pression d'air.
 - Le poids (W) est la force causée par la gravité.
- La portance (L) est la force qui maintient un avion en l'air. Les ailes créent la plus grande partie de la portance utilisée par le drone.

La façon dont les quatre forces agissent sur le drone définit son déplacement. Chaque force a une force opposée qui travaille contre elle. La portance fonctionne à l'opposé du poids. La poussée fonctionne à l'opposé de la traînée. Lorsque les forces sont équilibrées, le drone vole dans une direction horizontale (en croisière). Il monte si les forces de portance et de poussée sont supérieures au poids et à la traînée. Si le poids et la traînée sont plus grands que la portance et la poussée, le drone descend. Tout comme la traînée retient le drone en réponse à la force du vent, la portance le pousse vers le haut. La pression de l'air est plus élevée sur la partie inférieure de l'aile, donc elle est poussée vers le haut.

La figure 4.2 illustre ces forces qui agissent sur le drone en croisière où nous avons :

$$\begin{aligned}\vec{T} - \vec{D} &= 0 \\ \vec{L} - \vec{W} &= 0\end{aligned}\tag{4.1}$$

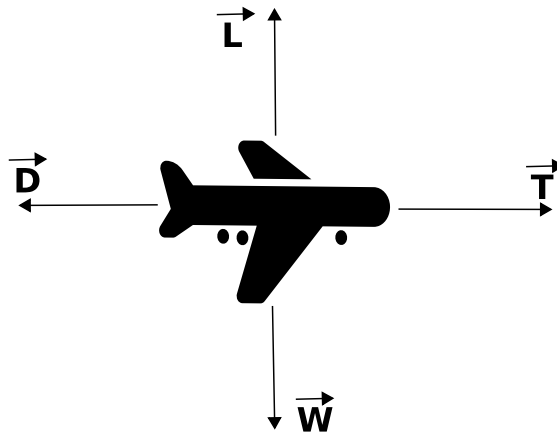


Figure 4.2 – Les forces qui agissent sur le drone en croisière.

Pour la montée un angle θ se forme entre l'axe horizontal et l'axe longitudinal du drone. Cet angle ne peut pas dépasser une valeur maximale θ_{max} qui est une caractéristique physique du drone. Dans ce cas les vecteurs de forces sont présentés dans la figure 4.3 de sorte que :

$$\begin{aligned}\vec{T} - \vec{D} - \vec{W} \sin \theta &= 0 \\ \vec{L} - \vec{W} \cos \theta &= 0\end{aligned}\tag{4.2}$$

Les mêmes équations sont utilisées pour la descente, mais avec un angle θ négatif.

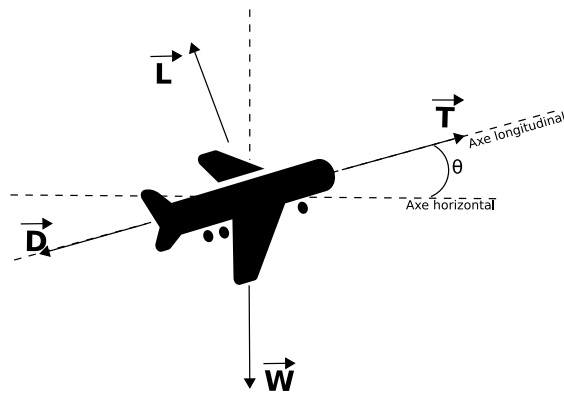


Figure 4.3 – Les forces qui agissent sur le drone en montée.

Pour tourner, un drone doit effectuer une inclinaison latérale. Dans ce cas, un angle Φ se forme entre l'axe horizontal et l'axe latéral et comme l'angle θ il ne doit pas dépasser une valeur maximale Φ_{max} . La figure 4.4 montre les forces exercées sur le drone en inclinaison latérale où :

$$\begin{aligned}\vec{T} - \vec{D} &= 0 \\ \vec{L} \cos \Phi - \vec{W} &= 0\end{aligned}\quad (4.3)$$

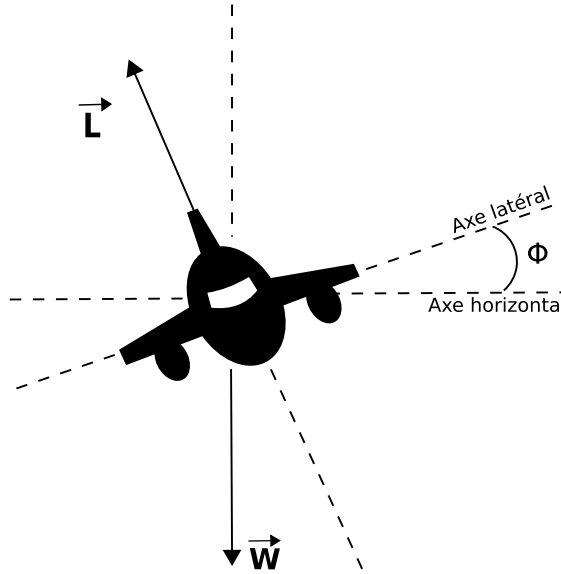


Figure 4.4 – Les forces qui agissent sur le drone en inclinaison latérale.

Le drone peut être en montée ou en descente et en même temps en inclinaison latérale. Nous avons donc les équations suivantes :

$$\begin{aligned}\vec{T} - \vec{D} - \vec{W} \sin \theta &= 0 \\ \vec{L} \cos \Phi - \vec{W} \cos \theta &= 0\end{aligned}\quad (4.4)$$

Le drone se déplace avec une vitesse V_R (figure 4.5) qui est la vitesse réelle du drone. Cette vitesse est en relation avec la vitesse relative au sol V_S et celle du vent V_V sous la forme d'une équation vectorielle comme suit :

$$\vec{V}_S = \vec{V}_R + \vec{V}_V\quad (4.5)$$

Le point qui représente le drone dans notre simulateur a toutes ces caractéristiques physiques. Cela permet à ce point (drone simulé) d'exécuter des actions élémentaires : monter, descendre

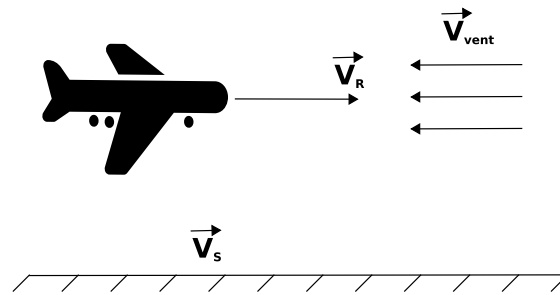


Figure 4.5 – La vitesse réelle du drone.

et tourner selon la façon dont les forces (T, D, W, L) agissent sur ce point.

Ces actions élémentaires permettent au drone simulé de naviguer à travers un ensemble de waypoints prédéfinis en avance. Mais il est aussi capable d'exécuter des actions de haut niveau plus complexes, par exemple retourner à la base, continuer, atterrir, sauter un waypoint, . . . L'exécution de ces actions de haut niveau est en réalité une succession ou une combinaison des actions élémentaires. Par exemple, si le drone décide de continuer la mission, cela revient en pratique à ralié le prochain waypoint.

Différentes situations sont possibles lors de l'exécution de ces actions de haut niveau. Elles dépendent de l'altitude du waypoint courant W_i et du waypoint cible W_{i+1} et l'angle entre les deux waypoints. En effet :

- Si W_i et W_{i+1} sont à la même altitude alors le drone simulé vole dans une direction horizontale.
- Si W_i est à une altitude plus basse que celle de W_{i+1} et $\theta \leq \theta_{max}$ alors le drone simulé monte directement vers W_{i+1}
- Si W_i est à une altitude plus basse que celle de W_{i+1} et $\theta > \theta_{max}$ alors le drone simulé ne peut pas monter directement vers W_{i+1} . Il monte en effectuant une spirale montante pour atteindre sa destination.
- Si W_i est à une altitude plus haute que celle de W_{i+1} et $\theta \geq -\theta_{max}$ alors le drone simulé descend directement vers W_{i+1}
- Si w_i est à une altitude plus haute que celle de W_{i+1} et $\theta < -\theta_{max}$ alors le drone simulé ne peut pas descendre directement vers W_{i+1} . Il descend en effectuant une spirale descendante pour atteindre sa destination.

4.3 Modélisation d'une mission

Une mission dans notre simulateur est modélisée par un ensemble de waypoints à survoler. Chaque waypoint est identifié par ses coordonnées tridimensionnelles : latitude, longitude et altitude. Un autre élément qui définit une mission est un ensemble d'objectifs à atteindre. Par exemple, en chaque waypoint une tâche pourrait être effectuée, ce qui contribuerait à atteindre un objectif.

Voler d'un waypoint à un autre nécessite de calculer une trajectoire. Puisque dans notre simulateur nous nous focalisons surtout sur le niveau 2, toutes les trajectoires sont des connexions directes entre les 2 waypoints tout en respectant les contraintes physiques (précisées ci-avant en section 4.2). Elles dépendent de l'exécution d'actions élémentaires (monter, descendre, tourner) et des altitudes des deux waypoints et de l'angle entre eux (θ).

Comme déjà mentionné, pour simplifier notre travail, nous remplaçons ici le niveau 1 par un système qui déclenche la prise de décision du niveau 2 à chaque waypoint. L'événement déclencheur est donc l'arrivée à un waypoint, qui déclenche donc une réévaluation du plan de mission, avec potentiellement un changement.

Lorsque le niveau 2 est déclenché, le simulateur génère la liste des actions possibles, en fonction du contexte du drone. Ce contexte est décrit par les contraintes physiques du drone, son état, son environnement et le type de mission qu'il est en train d'exécuter. Par exemple, les actions qui nécessitent plus de carburant que la quantité disponible pour les exécuter sont écartées de la liste. De la même manière, l'action « atterrir » n'est pas présente dans cette liste si le drone survole la mer.

Les calculs des évaluations des actions possibles sur les différents critères sont intégrés dans notre simulateur. Nous rappelons ici les critères choisis précédemment :

- le risque pour le drone, c'est-à-dire le risque associé au survol de différentes zones telles que forêts, mer, zones militaires,
- le risque pour l'environnement, tel que les personnes, les bâtiments, en cas d'accident, par exemple,
- la consommation d'énergie, correspondant à la quantité d'énergie restante après l'exécution de l'action sélectionnée,
- la réalisation de la mission, qui est un pourcentage pondéré des sous-objectifs atteints.

En général, il existe plusieurs façons pour modéliser le risque. Nous choisissons dans notre travail d'utiliser les données géographiques pour modéliser le risque pour le drone et le risque pour l'environnement. Nous générons donc deux cartes de risques pour la zone de vol du drone.

À une position donnée, la valeur de risque est obtenue en projetant la position du drone sur la carte de risque correspondante. Pour évaluer ce risque pour une action, nous calculons le risque pour chaque point de la trajectoire empruntée par le drone lors de l'exécution de cette action. Ensuite nous calculons la moyenne de ces risques pour donner l'évaluation du risque pour cette action.

Cette même méthode de calcul est utilisée pour l'évaluation risque pour l'environnement et du drone.

Pour l'évaluation des actions sur le troisième critère, un modèle de consommation d'énergie est utilisé. Ce modèle est détaillé dans la suite.

Pour calculer la consommation de carburant du drone, il faut connaître ses propriétés de vol. Dès qu'il quitte le sol, la distance qu'il parcourt en vol ne correspond pas à celle mesurée au sol. Deux raisons expliquent cette différence :

- La première est qu'un drone vole par rapport à la masse d'air qui l'entoure et qu'il peut être exposé au vent. La direction du vent influence la trajectoire de vol du drone, un vent qui vient de l'arrière augmente sa vitesse relative au sol tandis qu'un vent de face la diminue. Les vents latéraux introduisent un mouvement sur le côté qui détourne le drone de sa trajectoire s'il n'est pas géré.
- La deuxième raison est le fait qu'un drone ne vole pas seulement horizontalement, mais aussi verticalement pendant la montée et la descente dans un espace tridimensionnel.

Comme nous l'avons déjà présenté, la vitesse réelle \vec{V}_R du drone est en relation avec celle du vent \vec{V}_V et celle relative au sol \vec{V}_S , tel que $\vec{V}_S = \vec{V}_R + \vec{V}_V$.

Pour un segment j de la trajectoire de vol entre deux waypoints, le carburant nécessaire peut être calculé à l'aide de la formule suivante :

$$c_j = \frac{d_j \cdot TCC_j}{V_{S_j}} \quad (4.6)$$

avec, d_j la longueur du segment j , V_{S_j} la vitesse relative au sol et TCC_j le taux de consommation de carburant.

TCC_j est une relation entre la consommation du carburant et la puissance fournie par le moteur. Cette information fait partie des données non accessibles. Par conséquent, nous calculons les valeurs manquantes sur la base des données disponibles.

Les spécifications des avions commerciaux qui rentrent dans la même catégorie de consommation d'énergie que celle de « Thales Watchkeeper WK450 » ou une catégorie comparable peuvent aider à combler les lacunes avec des estimations raisonnables. Ceci est particulièrement vrai pour les différentes phases de vol comme le décollage, la montée, la croisière et la descente à différentes altitudes et vitesses.

Les équations liées aux forces qui agissent (le poids (W), la poussée (T), la portance (L) et la traînée (D)) sur le drone pour chaque mode de vol (en croisière, monter, descendre, tourner) ainsi que les informations à notre disposition permettent de calculer les paramètres nécessaires pour estimer la consommation d'énergie entre deux waypoints.

À cette fin, la trajectoire entre 2 waypoints est divisée en n segments, et le calcul de la consommation se fait en utilisant pour chaque segment l'équation 4.6. La raison principale pour laquelle la trajectoire est divisée est l'influence du poids et de l'altitude à laquelle le drone vole sur la consommation d'énergie. En effet, pour une montée ou une descente la densité d'air change de façon significative ce qui à son tour influence le besoin en puissance et donc la consommation de carburant.

Ensuite la consommation finale est obtenue en sommant les consommations de chaque segment. Pour chaque segment, la consommation de carburant est calculée en fonction de l'altitude (la densité de l'air), le mode de vol (en croisière, en montée/descente ou en inclinaison latérale) et d'autres influences physiques. Dans notre cas, nous supposons que la vitesse est constante de sorte que les temps d'accélération sont très faibles par rapport au temps de vol. Nous décidons donc de négliger les phases d'accélération pour les calculs dans notre simulateur.

Cela conduit à l'équation de consommation de carburant suivante pour l'ensemble de la trajectoire de vol entre deux waypoints W_i et W_{i+1} :

$$c_{W_i, W_{i+1}} = \sum_{j=1}^n \frac{d_j * TCC_j}{V_{S_j}} = \sum_{j=1}^n TCC_j t_j \quad (4.7)$$

où t_j est le temps de vol pour le segment j considéré.

Ceci peut également se noter comme une intégrale en fonction du temps :

$$c_{W_i, W_{i+1}} = \int_{t_i}^{t_{i+1}} TCC dt \quad (4.8)$$

où TCC est une représente le taux de consommation de carburant en fonction du temps.

Le taux de consommation de carburant du drone dépend des manœuvres de vol et de la durée du vol. Dans notre cas nous supposons que la puissance du moteur est proportionnelle à la quantité de carburant. L'apport fourni est de $\dot{m}_c = 11.8kg/h$ à 70% de la puissance du moteur (qui est utilisé comme puissance de référence ici).

Pour distinguer l'influence des différentes manœuvres de vol, un facteur d'efficacité est introduit. Il fait le rapport entre la puissance requise P_R pour un mode de vol et de la puissance motrice disponible P_A .

$$\eta_c = \frac{P_R}{P_A} \quad (4.9)$$

Le taux de consommation de carburant en kg/s pour un mode de vol spécifique peut être calculé par :

$$TCC = \dot{m}_c \eta_c \quad (4.10)$$

Cela nous mène à calculer la consommation du carburant en vol par :

$$c_{W_i, W_{i+1}} = \int_{t_i}^{t_{i+1}} \dot{m}_c \eta_c dt \quad (4.11)$$

où le facteur d'efficacité η_c n'est pas constant, mais dépend du temps.

Pour terminer, concernant le critère de la réalisation de la mission, nous le calculons simplement comme un pourcentage exprimant la proportion d'objectifs atteints (par rapport aux objectifs initiaux) si l'action est exécutée.

Une fois que toutes les actions possibles sont évaluées sur les différents critères, un modèle SRMP est exécuté pour ordonner les différentes actions possibles du drone, et lui permettre ainsi de prendre une décision (comme expliqué dans le chapitre 3). Les paramètres de ce modèle sont fixés avant de l'intégrer dans le simulateur de sorte qu'ils représentent au mieux les préférences de l'opérateur. Les méthodes d'élicitation des paramètres d'un modèle SRMP seront détaillées

dans la troisième partie de la thèse.

Le simulateur offre la possibilité à l'utilisateur de mettre en pause l'exécution d'une mission afin de se concentrer sur certains détails comme les évaluations des différentes actions ou d'analyser la réalisation de la simulation ou même pouvoir valider ou non les décisions prises par le drone.

4.4 Interface utilisateur

L'IHM (interface homme-machine) du simulateur présentée dans la figure 4.6 est composée de quatre parties principales. La fenêtre à gauche est réservée à une représentation synthétique de la mission. Nous considérons ici que la mission est planifiée à l'avance c'est-à-dire que l'ensemble des waypoints, la zone de vol et les cartes de risque sont déjà fixés.

Le simulateur affiche la zone de vol en 3D ce qui permet de visualiser l'environnement selon n'importe quel point de vue. Les différents waypoints qui spécifient les différentes étapes de la mission sont représentés en croix rouges selon leur position définie par les coordonnées géographiques : latitude, longitude et altitude. Les axes affichés dans la fenêtre représentent donc des axes de latitude, de longitude et d'altitude.

La carte géographique de la zone de vol est également affichée. Son but est d'informer l'utilisateur de la localisation du drone. Comme mentionné précédemment, nous avons choisi de représenter le risque pour l'environnement et le risque pour le drone par deux autres données géographiques. Ces cartes de risque sont affichées de manière superposées, et sont composées de différentes géométries comportant différentes valeurs de risque (représentées par des couleurs différentes).

Le drone est représenté par un point bleu à chaque moment de la mission, ce qui laisse une trace sur sa trajectoire déjà parcourue représentée par l'ensemble des points de passage bleus. Lorsque les points de passage sont rouges, cela signifie que le vol ne peut pas s'effectuer en direct, mais nécessite un vol en spirale (la valeur absolue de l'angle entre les 2 waypoints est supérieure à ce qui est toléré par le drone).

La fenêtre en haut à droite fournit les informations sur la position actuelle du drone et la position de sa prochaine destination selon le plan de la mission en affichant leurs coordonnées géographiques (latitude, longitude et altitude). Des informations liées à l'état du drone sont aussi affichées, sa vitesse actuelle en m/s ainsi que la quantité de carburant restante en kg .

La fenêtre du milieu à droite présente l'historique des actions exécutées par le drone jusqu'à l'instant actuel. Ce tableau permet d'afficher les détails des évaluations de ces actions sur les différents critères.

Lorsque le drone est face à une décision de haut niveau (c'est-à-dire que le niveau 2 du moteur de décisions du chapitre 3 a été déclenché par l'arrivée dans un waypoint), l'ensemble des actions possibles (atterrir, continuer, ...) et les valeurs de leurs évaluations sur les différents critères (risque pour le drone, risque pour l'environnement, la consommation d'énergie, la réalisation de la mission) sont affichées dans la fenêtre en bas à droite. L'action recommandée par le moteur de décisions est alors surlignée en vert et le drone la met en oeuvre. L'opérateur est donc informé

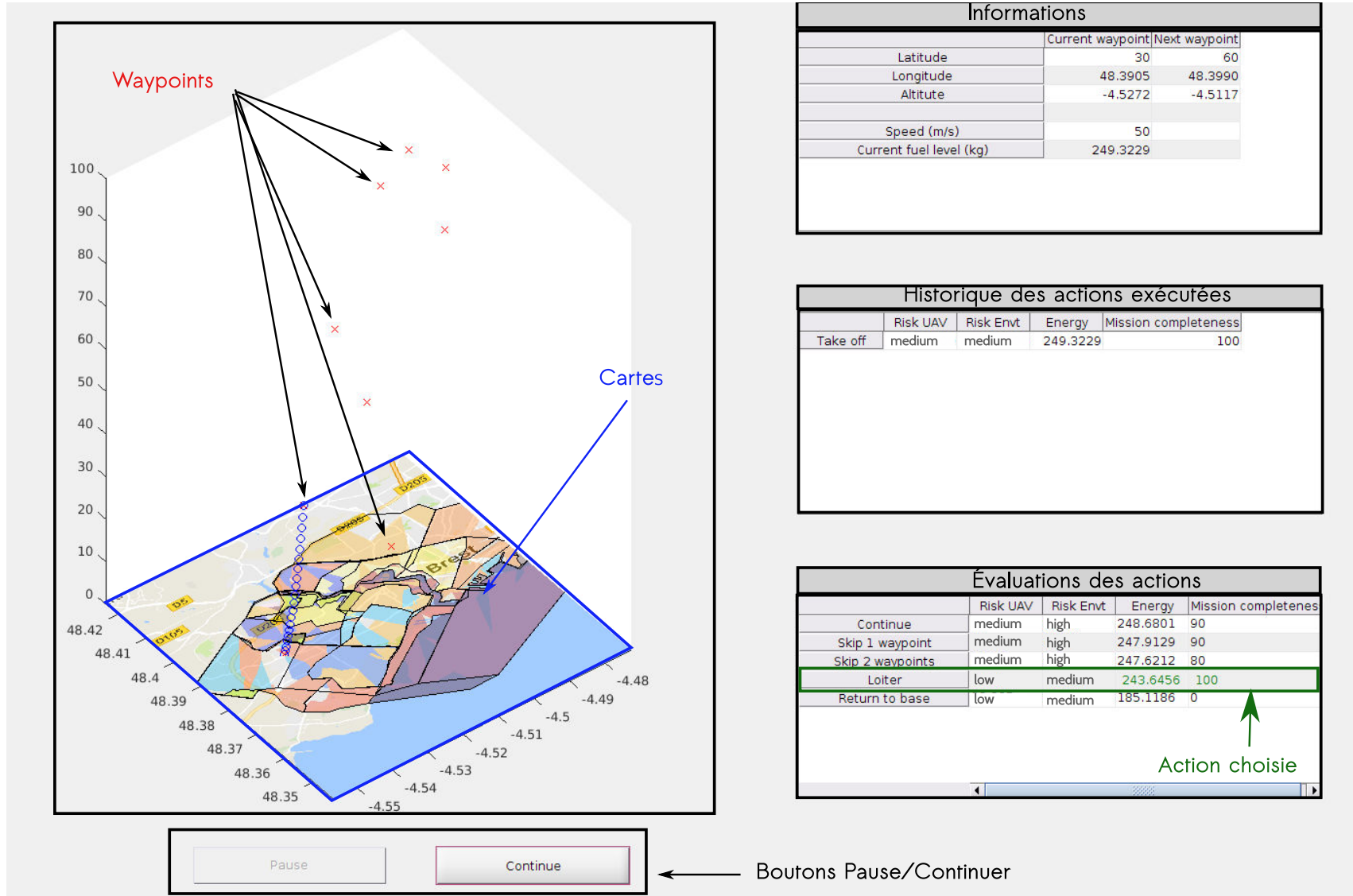


Figure 4.6 – Moteur de décisions multi-niveaux

de la décision prise par le drone.

4.5 Exemple d'exécution de mission

Comme déjà mentionné au début ce chapitre le but de développer ce simulateur est d'illustrer l'intérêt de notre moteur de décisions multi-niveaux. Plus particulièrement, nous souhaitons montrer l'intérêt de l'intégration des préférences d'un opérateur dans la prise de décision du drone.

Nous illustrons donc le fonctionnement du simulateur dans cette section, en soulignant l'apport du moteur de décisions proposé au chapitre 3, et plus particulièrement l'intégration de préférences dans le niveau 2. Afin de bien souligner l'intérêt, nous utilisons deux opérateurs dont les modèles de préférences SRMP sont distincts. Nous observons alors dans le simulateur l'effet de ces deux modèles sur la réalisation de la mission.

4.5.1 Exemple de mission

Nous présentons ici un exemple de mission, où le drone doit accomplir une mission qui consiste à survoler la ville de Brest en France à travers un ensemble de huit waypoints et prendre des photos en chaque point et retourner à sa position initiale. De plus, pour les waypoints 1, 5 et 6, deux photos sont nécessaires, ce qui oblige le drone à faire un tour pour une deuxième prise. Pour rappel, dans la version actuelle du simulateur, la décision de niveau 2 du moteur de décisions est déclenchée automatiquement à chaque waypoint. Deux cartes des risques liées à la zone de vol sont fournies, la première représentant le risque pour le drone et la seconde le risque pour l'environnement en cas d'accident. Nous exécutons la mission selon deux profils d'opérateur différents, représentés par deux ensembles différents de paramètres de préférences SRMP.

Le tableau 4.2 présente les coordonnées géographiques de l'ensemble des waypoints.

Waypoints	Latitude	Longitude	Altitude
$W_{initial}$	48.387266	-4.535294	0
W_1	48.390458	-4.527225	30
W_2	48.399007	-4.51169	60
W_3	48.397126	-4.500275	90
W_4	48.391655	-4.496584	100
W_5	48.382136	-4.493752	100
W_6	48.377062	-4.497957	90
W_7	48.37558	-4.521217	60
W_8	48.367997	-4.520187	30

Tableau 4.2 – Les coordonnées géographiques des waypoints

Les deux cartes de risque pour le drone et pour l'environnement sont réalisées à l'aide de la plateforme *deSEAsion* [Shom et IMT Atlantique, 2017]. Cette plateforme permet de réaliser des évaluations qualitatives ou quantitatives des zones géographiques, en agrégeant des données brutes. Elle intègre la perspective humaine (des acteurs de la prise de décision) dans cette évaluation en utilisant des modèles de décision (base de règles, modèles d'AMCD, ...) en générant des cartes adaptées aux perspectives des décideurs.

Pour construire la carte de risque pour le drone, nous avons utilisé la base de données CORINE Land Cover (CLC) 2012 [Agence européenne pour l'environnement, 2012] qui est une base de données géographique d'occupation des sols. Elle fournit les types des zones qui sont répartis sur 5 catégories :

1. Territoires artificialisés, qui contiennent les zones urbaines, industrielles, les mines et les espaces verts artificialisés.
2. Territoires agricoles qui incluent les terres arables, les cultures permanentes, les prairies et les zones agricoles hétérogènes.
3. Forêts et milieux semi-naturels, qui contiennent les forêts, les milieux à végétations, et les espaces ouverts.
4. Zones humides intérieures et maritimes (les marais).
5. Surfaces d'eau, qui incluent les eaux continentales et maritimes.

L'évaluation du risque pour le drone dans une zone de vol est donc obtenue à l'aide de l'utilisation de la plateforme *deSEAsion* qui prend comme entrée la base de données géographique CLC2012 et l'ensemble des règles suivant :

- Si le drone survole une zone de type agricole alors le risque est faible,
- Si le drone survole une zone de type forêt alors le risque est moyen,
- Si le drone survole une zone humide alors le risque est haut,
- Si le drone survole un surface d'eau alors le risque est très haut.

DESEASION permet également de découper une carte en fonction de la zone de vol considérée (dans notre cas, la ville de Brest). Cela a un avantage pour notre simulateur parce qu'il y a moins de données géographiques à traiter et donc nous permet de gagner en temps d'exécution.

La figure 4.7 montre la carte générée pour le risque pour le drone. Chaque zone est soit évaluée « faible », « moyen », « haut » ou « très haut ». Les zones en rouge sont les zones les plus risquées alors que celles en bleu ciel se sont les moins risquées. Nous remarquons que cette carte respecte les règles définies précédemment : la zone avec le risque le plus élevé (en rouge) pour le drone en cas d'accident est la mer, alors que le niveau le plus bas (en bleu ciel) est sur les zones agricoles ou les zones situées en dehors de la ville.

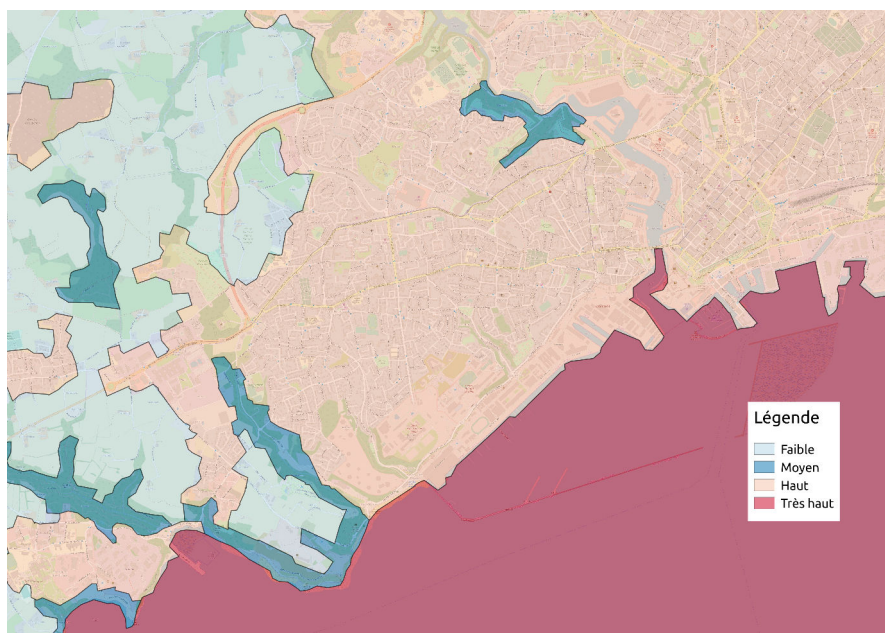


Figure 4.7 – Carte de risque pour le drone.

Pour la carte de risque pour l'environnement, deux bases de données géographiques sont utilisées pour sa génération. La première est celle utilisée pour la construction de la carte de risque pour le drone, la CORINE Land Cover2012. La deuxième est la base de données d'estimation carroyées de population [INSEE, 2010] proposée par l'Institut National de la Statistique et des Études Économiques (INSEE). Cette base contient l'estimation de nombre d'habitants par un carreau de 200 par 200 mètres.

L'utilisation de la plateforme *deSEAsion* permet de fusionner ces données pour générer une seule carte en utilisant la base de règles suivante :

- Si le drone survole une surface d'eau alors le risque est faible,
- Si le drone survole une forêt ou une zone humide alors le risque est moyen,
- Si le drone survole une zone industrielle et moyennement peuplée (< 10 habitants) alors le risque est haut,
- Si le drone survole une zone urbaine et peuplée (≥ 10 habitants) alors le risque est très haut.

Dans la figure 4.8, nous montrons la carte de risque pour l'environnement. Nous prenons ici la même échelle où les zones en rouge sont les zones les plus risquées alors que celles en bleu ciel sont les moins risquées. Cette carte montre bien qu'en cas d'accident dans une zone peuplée (la zone en rouge) le risque pour l'environnement est plus élevé. Ici la densité de la population de chaque zone est prise en compte pour calculer le niveau du risque.

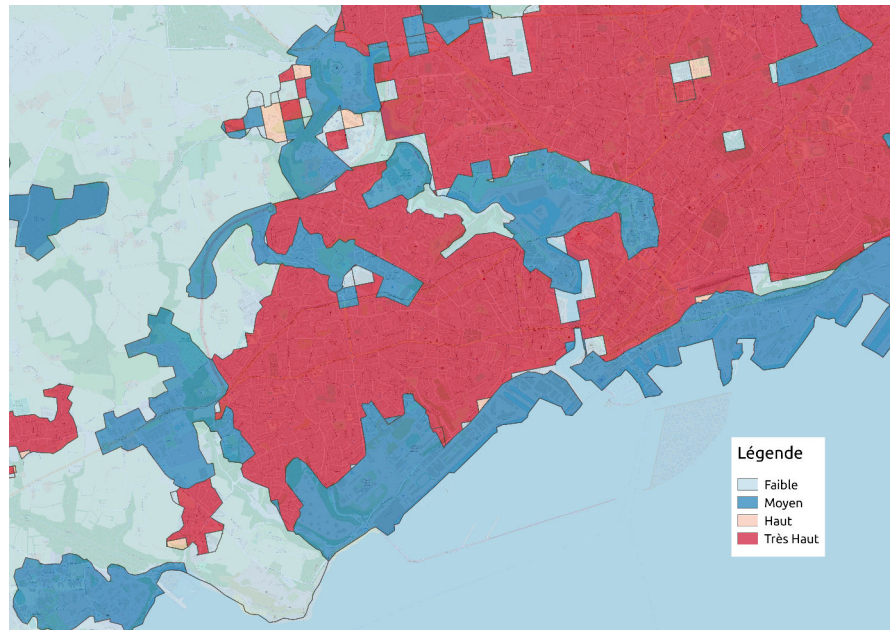


Figure 4.8 – Carte de risque pour l'environnement.

Nous considérons deux opérateurs avec deux profils différents où le premier se concentre principalement sur l'achèvement de la mission, en plaçant comme objectifs secondaires le risque et la consommation d'énergie. Ce profil est traduit en paramètres de préférence pour un modèle SRMP présentés dans le tableau 4.3 et la figure 4.9.

		R_{UAV}	R_{Env}	E	M
Opérateur #1	p^1	haut	très haut	30%	30%
	p^2	faible	moyen	60%	90%
	w	0.1	0.1	0.1	0.7
	σ		(2, 1)		

Tableau 4.3 – Paramètres de préférences du premier opérateur.

Le deuxième opérateur tient également compte des résultats de la mission, mais en même temps, il veut être prudent et ne pas dépasser un certain niveau de risque. Les paramètres de préférences représentant cet opérateur sont résumés dans le tableau 4.4 et la figure 4.10.

Ces paramètres préférentiels sont typiquement appris par une phase d'élicitation des préférences. Nous reviendrons sur cet aspect dans la troisième partie de notre travail.

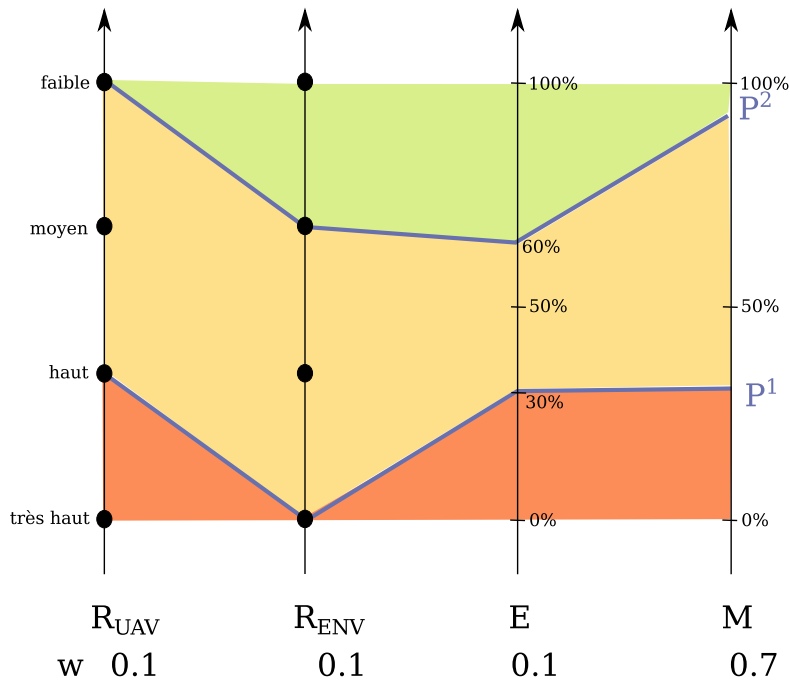


Figure 4.9 – Modèle SRMP de l'opérateur #1

		R_{UAV}	R_{Env}	E	M
Opérateur #2	p^1	haut	très haut	30%	30%
	p^2	faible	moyen	60%	70%
	w	0.25	0.25	0.25	0.25
	σ		(2, 1)		

Tableau 4.4 – Paramètres de préférences du deuxième opérateur.

4.5.2 Exécution de la mission dans le simulateur

La figure 4.11 montre l'exécution de la mission avec les préférences du premier opérateur. Sachant que le drone prend une décision de haut niveau à chaque waypoint, nous constatons qu'il est passé par tous les waypoints prévus et il a même effectué un tour autour des waypoints 1, 5 et 6 pour reprendre les photos (montré par le cercle bleu dans la figure) sans tenir compte du risque lié aux zones de vol (zone rouge sur la carte risque).

La figure 4.12 montre l'exécution de la mission en prenant en considération les préférences du deuxième opérateur. Nous remarquons que le drone prend bien deux photos dans le waypoint 1, pour le waypoint 5 il prend une seule photo parce qu'il estime que survoler de nouveau la même zone est trop risqué. Nous remarquons aussi que drone saute le waypoint 6 pour la même raison.

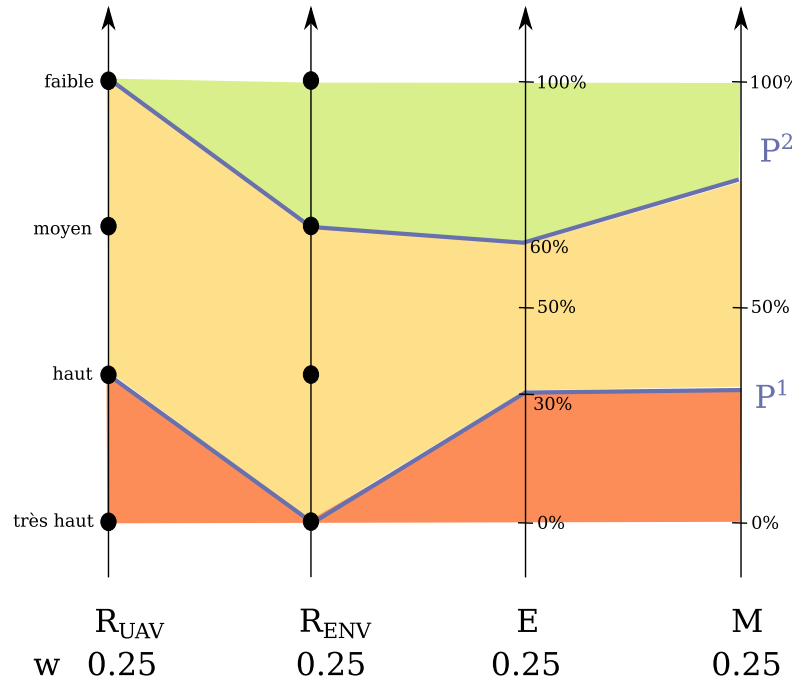


Figure 4.10 – Modèle SRMP de l'opérateur #2

Comme déjà mentionné auparavant l'une des raisons pour laquelle notre choix est tombé sur le modèle SRMP est la facilité d'expliquer les recommandations. En nous basant sur cette caractéristique, nous expliquons dans la suite quelques décisions prises par le drone selon chacun des profils des deux opérateurs dans de différentes étapes de la mission.

Nous prenons comme premier exemple le comportement du drone dans le waypoint 1 selon les modèles SRMP des deux opérateurs. Au waypoint 1, cinq actions sont possibles et elles sont évaluées comme montré dans le tableau 4.5.

	R_{UAV}	R_{Env}	E	M
continuer (C)	faible	haut	99%	90%
sauter 1 waypoint (S1W)	moyen	haut	95%	80%
sauter 2 waypoint (S2W)	moyen	haut	93%	70%
faire un tour (L)	faible	moyen	96%	100%
retourner à la base (RB)	faible	moyen	97%	10%

Tableau 4.5 – Évaluation des actions au waypoint 1.

Pour classer les alternatives selon le modèle SRMP de l'opérateur #1 qui sont représentées dans la figure 4.13, nous comparons d'abord chacune des alternatives aux autres à travers les

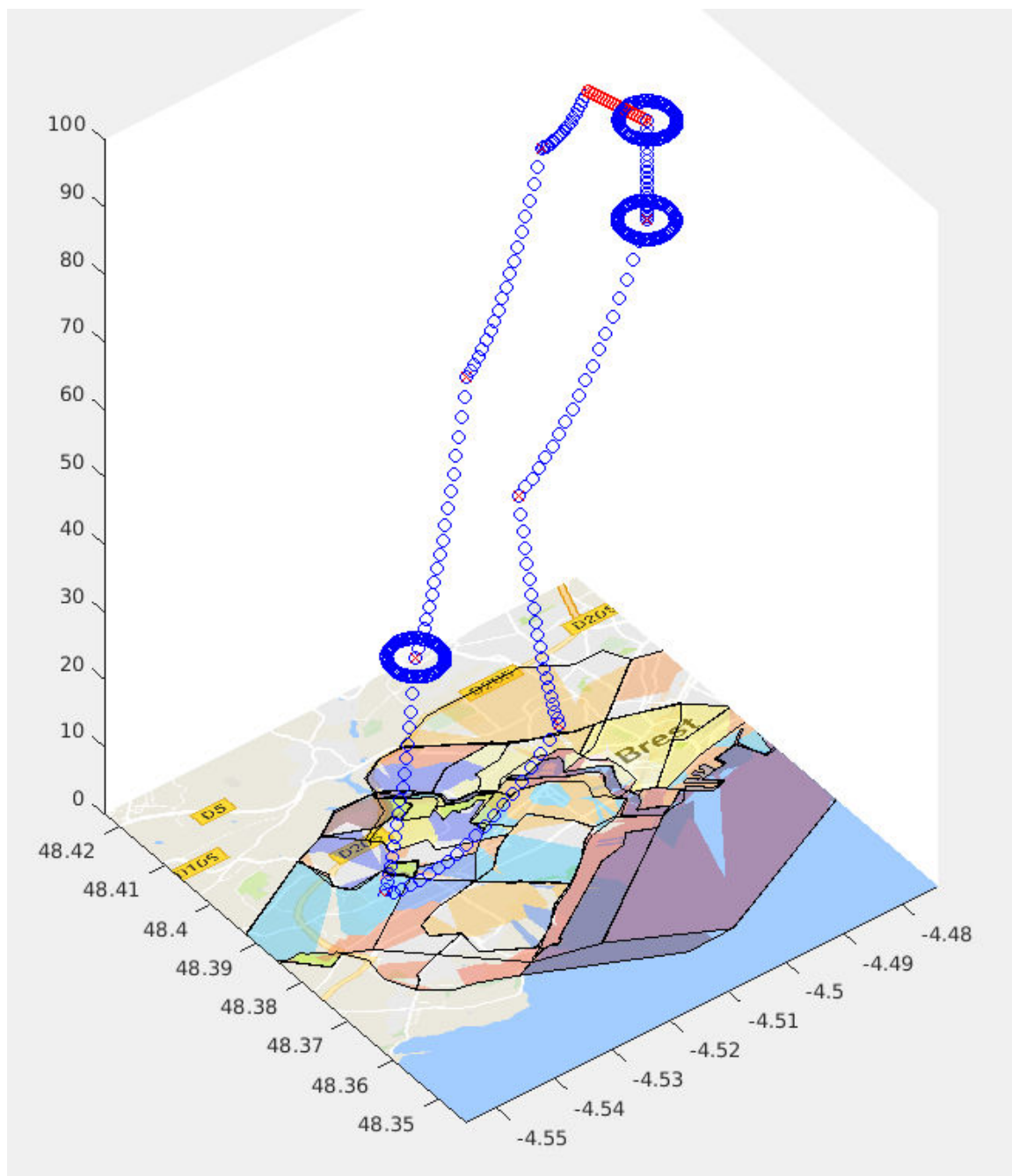


Figure 4.11 – Simulation de la mission avec les préférences de l'opérateur #1

profils dans l'ordre lexicographique $\sigma = (2, 1)$.

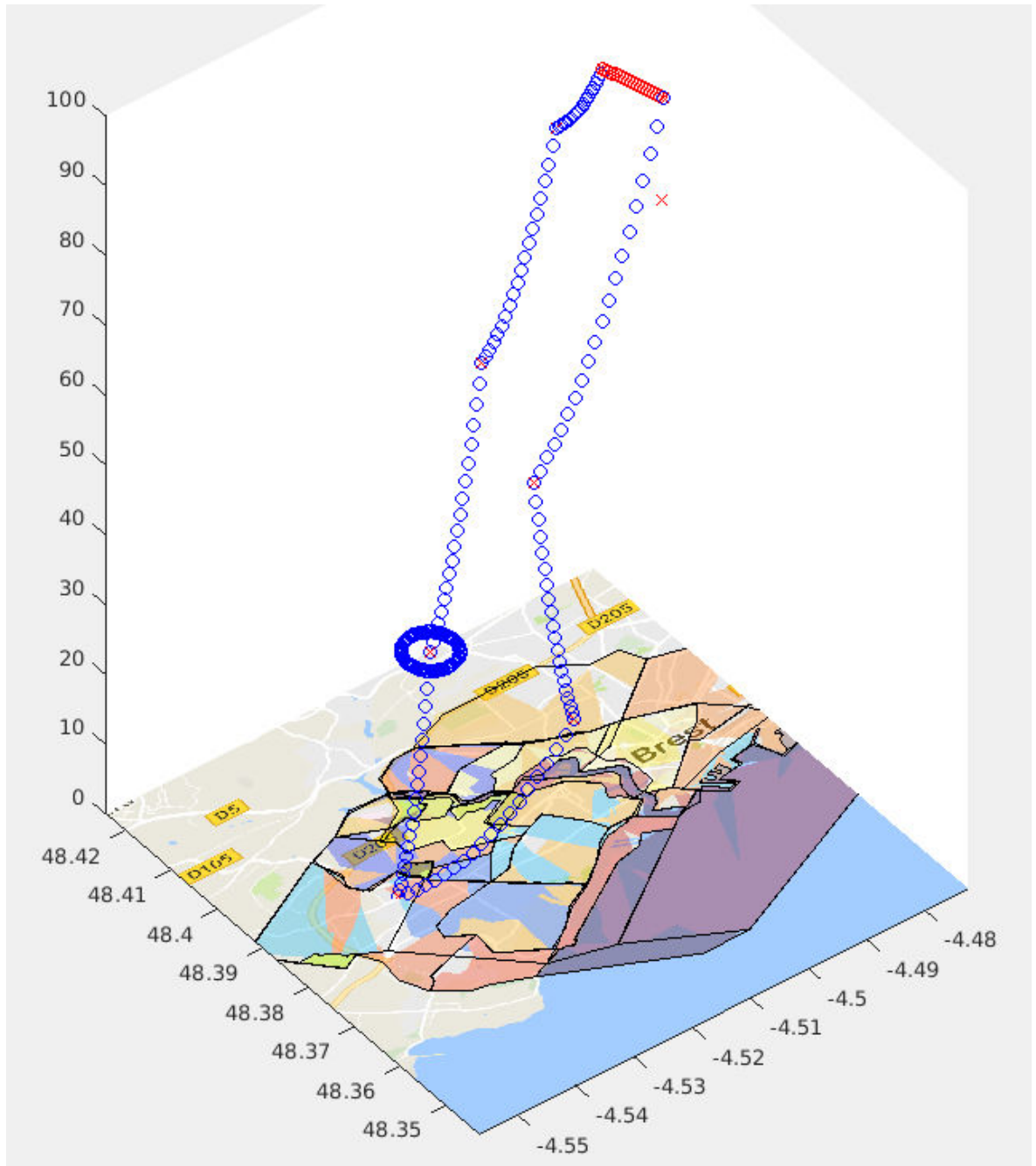


Figure 4.12 – Simulation de la mission avec les préférences de l'opérateur #2

Pour p^2 :

$$\left. \begin{array}{l} \sum_{j \in C(C,p^2)} w_j = 0.1 + 0 + 0.1 + 0.7 = 0.9 \\ \sum_{j \in C(S1W,p^2)} w_j = 0 + 0 + 0.1 + 0 = 0.1 \\ \sum_{j \in C(S2W,p^2)} w_j = 0 + 0 + 0.1 + 0 = 0.1 \\ \sum_{j \in C(L,p^2)} w_j = 0.1 + 0.1 + 0.1 + 0.7 = 1 \\ \sum_{j \in C(RB,p^2)} w_j = 0.1 + 0.1 + 0.1 + 0 = 0.3 \end{array} \right\} \Rightarrow \begin{array}{lll} L \succ_{p^2} C & C \succ_{p^2} S1W & RB \succ_{p^2} S2W \\ L \succ_{p^2} S1W & C \succ_{p^2} S2W & S1W \sim_{p^2} S2W \\ L \succ_{p^2} S2W & C \succ_{p^2} RB & \\ L \succ_{p^2} RB & RB \succ_{p^2} S1W & \end{array}$$

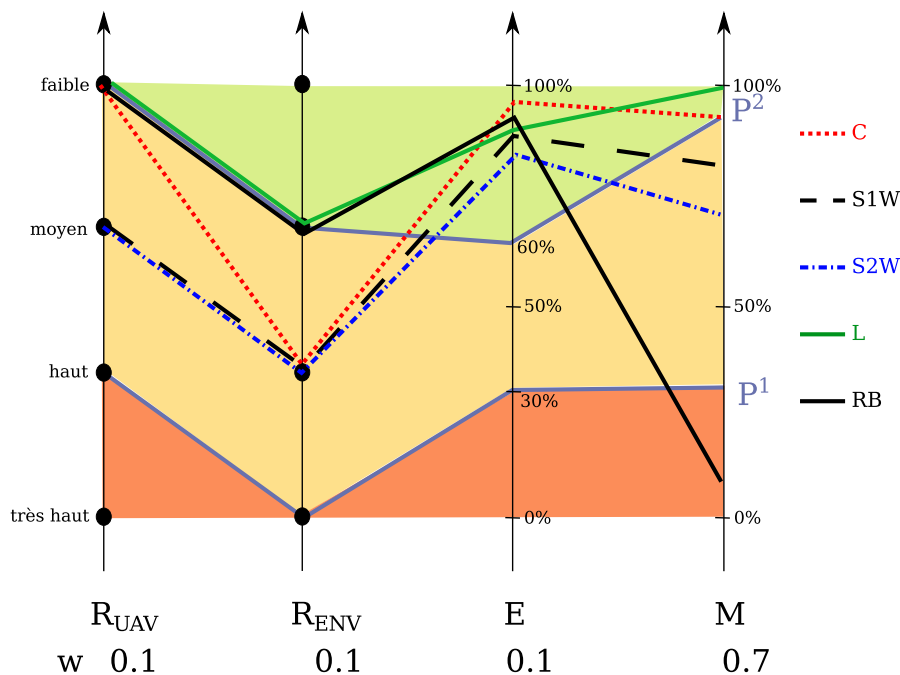


Figure 4.13 – Représentation des 5 actions dans le modèle SRMP de l'opérateur #1

Le profil p^2 n'est pas capable de discriminer entre $S1W$ et $S2W$, et donc nous continuons avec le profil p^1 , en respectant l'ordre lexicographique.

Pour p^1 :

$$\left. \begin{array}{l} \sum_{j \in C(S1W, p^1)} w_j = 0.1 + 0.1 + 0.1 + 0.7 = 1 \\ \sum_{j \in C(S2W, p^1)} w_j = 0.1 + 0.1 + 0.1 + 0.7 = 1 \end{array} \right\} \Rightarrow S1W \sim_{p^1} S2W$$

Le classement final est ainsi $L \succ C \succ RB \succ S1W \sim S2W$, donc L est considéré globalement comme la meilleure alternative. En d'autres termes, le drone choisira l'action L , alors que si un facteur externe l'en empêche (panne, apparition d'obstacle, ...), il choisira l'action C et ainsi de suite. Mais pendant cette simulation rien aucun événement n'a empêché l'action recommandée, et donc le drone fait un tour pour reprendre la deuxième photo.

Nous rappelons que cet opérateur se concentre principalement sur l'achèvement de la mission. Nous remarquons bien que le drone en choisissant l'action L respecte les préférences de cet opérateur puisque cette action offre 100% de la réalisation de la mission. Même en cas d'empêchement d'exécution de cette action, l'exécution de l'action continuer (C) garantit 90% de la réalisation de la mission.

Pour le deuxième opérateur, les alternatives sont représentées dans la figure 4.14.

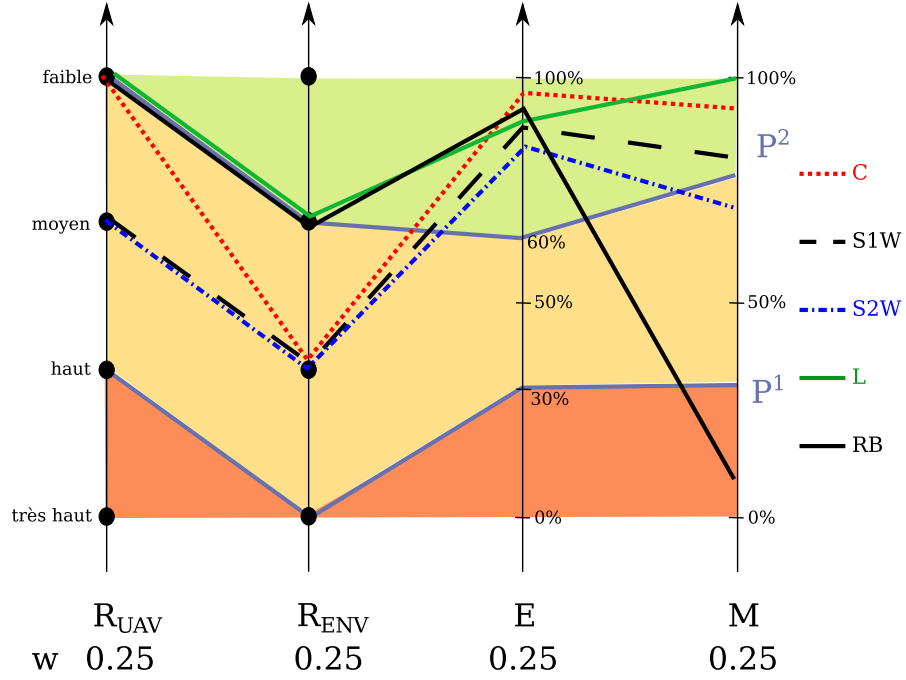


Figure 4.14 – Représentation des 5 actions dans le modèle SRMP de l'opérateur #2

Nous détaillons les calculs pour déterminer le classement des alternatives dans le waypoint 1 selon le profil de l'opérateur #2 ci-après.

Pour p^2 :

$$\left. \begin{array}{l}
 \sum_{j \in C(C, p^2)} w_j = 0.25 + 0 + 0.25 + 0.25 = 0.75 \\
 \sum_{j \in C(S1W, p^2)} w_j = 0 + 0 + 0.25 + 0.25 = 0.5 \\
 \sum_{j \in C(S2W, p^2)} w_j = 0 + 0 + 0.25 + 0.25 = 0.5 \\
 \sum_{j \in C(L, p^2)} w_j = 0.25 + 0.25 + 0.25 + 0.25 = 1 \\
 \sum_{j \in C(RB, p^2)} w_j = 0.25 + 0.25 + 0.25 + 0 = 0.75
 \end{array} \right\} \Rightarrow \begin{array}{lll}
 L \succ_{p^2} C & C \succ_{p^2} S1W & RB \succ_{p^2} S2W \\
 L \succ_{p^2} S1W & C \succ_{p^2} S2W & S1W \sim_{p^2} S2W \\
 L \succ_{p^2} S2W & C \sim_{p^2} RB & \\
 L \succ_{p^2} RB & RB \succ_{p^2} S1W &
 \end{array}$$

Le profil p^2 n'est pas capable de discriminer entre $S1W$ et $S2W$ et entre C et RB , et donc nous continuons avec le profil p^1 , en respectant l'ordre lexicographique.

Pour p^1 :

$$\left. \begin{array}{l}
 \sum_{j \in C(C, p^1)} w_j = 0.25 + 0.25 + 0.25 + 0.25 = 1 \\
 \sum_{j \in C(RB, p^1)} w_j = 0.25 + 0.25 + 0.25 + 0 = 0.75 \\
 \sum_{j \in C(S1W, p^1)} w_j = 0.25 + 0.25 + 0.25 + 0.25 = 1 \\
 \sum_{j \in C(S2W, p^1)} w_j = 0.25 + 0.25 + 0.25 + 0.25 = 1
 \end{array} \right\} \Rightarrow \begin{array}{l}
 C \succ_{p^1} RB \\
 S1W \sim_{p^1} S2W
 \end{array}$$

Le classement final est ainsi $L \succ C \succ RB \succ S1W \sim S2W$, donc le drone choisira aussi de faire un tour pour reprendre une deuxième photo comme pour l'opérateur #1.

Cet opérateur accorde aussi une grande importance à la réalisation de la mission, mais en plus il est prudent et donne une importance au niveau de risque. Donc suite à cette simulation, nous constatons que le drone respecte bien ces priorités. En effet le choix de l'action « faire un tour » (L) assure 100% de la réalisation de la mission et en plus avec un risque de drone « faible » et un risque d'environnement moyen, ce qui est en accord avec les priorités de l'opérateur #2. Mais aussi nous remarquons que cette action est meilleure que la suivante, continuer (C), selon ses priorités puisque cette dernière n'assure que 90% de la réalisation de la mission et elle est évaluée comme faible sur le risque de drone, mais haut sur le risque d'environnement.

Nous prenons le cas où le drone est au waypoint 5 comme un autre exemple pour expliquer ses décisions selon les deux profils des opérateurs. Dans le waypoint 5 cinq actions sont possibles. Dans le tableau 4.6 nous présentons leurs évaluations sur chaque critère.

	R_{UAV}	R_{Env}	E	M
continuer (C)	très haut	moyen	69%	90%
sauter 1 waypoint (S1W)	haut	moyen	65%	80%
sauter 2 waypoint(S2W)	haut	moyen	55%	70%
faire un tour (L)	très haut	moyen	76%	100%
retourner à la base(RB)	haut	moyen	49%	60%

Tableau 4.6 – Évaluation des actions au waypoint 5.

Ces actions sont représentées dans la figure 4.15 pour le premier opérateur.

Nous utilisons le modèle SRMP de l'opérateur #1 pour classer les actions selon ses préférences et justifier le choix du drone. Nous déroulons ci-dessous l'exécution du modèle SRMP :

Pour p^2 :

$$\left. \begin{array}{l} \sum_{j \in C(C, p^2)} w_j = 0 + 0.1 + 0.1 + 0 = 0.2 \\ \sum_{j \in C(S1W, p^2)} w_j = 0 + 0.1 + 0.1 + 0 = 0.2 \\ \sum_{j \in C(S2W, p^2)} w_j = 0 + 0.1 + 0 + 0 = 0.1 \\ \sum_{j \in C(L, p^2)} w_j = 0 + 0.1 + 0.1 + 0.7 = 0.9 \\ \sum_{j \in C(RB, p^2)} w_j = 0 + 0.1 + 0 + 0 = 0.1 \end{array} \right\} \Rightarrow \begin{array}{lll} L \succ_{p^2} C & C \sim_{p^2} S1W & S1W \succ_{p^2} S2W \\ L \succ_{p^2} S1W & C \succ_{p^2} S2W & S1W \succ_{p^2} RB \\ L \succ_{p^2} S2W & C \succ_{p^2} RB & \\ L \succ_{p^2} RB & RB \sim_{p^2} S2W & \end{array}$$

Le profil p^2 n'est pas capable de discriminer entre RB et $S2W$ et entre C et $S1W$, et donc nous continuons avec le profil p^1 , en respectant l'ordre lexicographique.

Pour p^1 :

$$\left. \begin{array}{l} \sum_{j \in C(C, p^1)} w_j = 0 + 0.1 + 0.1 + 0.7 = 0.9 \\ \sum_{j \in C(S1W, p^1)} w_j = 0.1 + 0.1 + 0.1 + 0.7 = 1 \\ \sum_{j \in C(S2W, p^1)} w_j = 0.1 + 0.1 + 0.1 + 0.7 = 1 \\ \sum_{j \in C(RB, p^1)} w_j = 0.1 + 0.1 + 0.1 + 0.7 = 1 \end{array} \right\} \Rightarrow \begin{array}{l} S1W \succ_{p^1} C \\ RB \sim_{p^1} S2W \end{array}$$

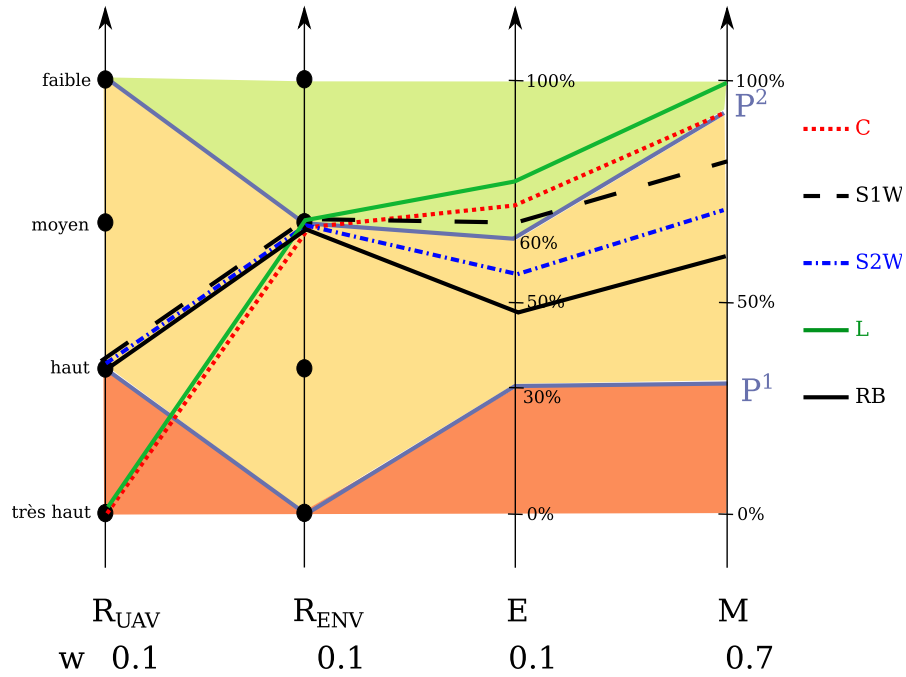


Figure 4.15 – Représentation des 5 actions dans le modèle SRMP de l'opérateur #1

Le classement final est $L \succ S1W \succ C \succ RB \sim S2W$, donc L est choisi par le drone. Pendant cette simulation le drone fait donc un tour pour reprendre une deuxième photo. Cela respecte les priorités de cet opérateur puisqu'il concentre plus sur la réalisation de la mission.

Ces actions sont représentées dans la figure 4.16 pour le premier opérateur.

Pour le deuxième opérateur, l'exécution du modèle SRMP se déroule comme suit :

Pour p^2 :

$$\left. \begin{array}{l} \sum_{j \in C(C, p^2)} w_j = 0.25 + 0 + 0.25 + 0.25 = 0.75 \\ \sum_{j \in C(S1W, p^2)} w_j = 0 + 0.25 + 0.25 + 0.25 = 0.75 \\ \sum_{j \in C(S2W, p^2)} w_j = 0 + 0.25 + 0 + 0.25 = 0.5 \\ \sum_{j \in C(L, p^2)} w_j = 0 + 0.25 + 0.25 + 0.25 = 0.75 \\ \sum_{j \in C(RB, p^2)} w_j = 0 + 0.25 + 0 + 0 = 0.25 \end{array} \right\} \Rightarrow \begin{array}{lll} L \sim_{p^2} C & C \sim_{p^2} S1W & S1W \succ_{p^2} S2W \\ L \sim_{p^2} S1W & C \succ_{p^2} S2W & S2W \sim_{p^2} RB \\ L \succ_{p^2} S2W & C \succ_{p^2} RB & \\ L \succ_{p^2} RB & S1W \succ_{p^2} RB & \end{array}$$

Le profil p^2 n'est pas capable de discriminer entre L et SW et C , et donc nous continuons avec le profil p^1 .

Pour p^1 :

$$\left. \begin{array}{l} \sum_{j \in C(C, p^1)} w_j = 0 + 0.25 + 0.25 + 0.25 = 0.75 \\ \sum_{j \in C(S1W, p^1)} w_j = 0.25 + 0.25 + 0.25 + 0.25 = 1 \\ \sum_{j \in C(L, p^1)} w_j = 0 + 0.25 + 0.25 + 0.25 = 0.75 \end{array} \right\} \Rightarrow \begin{array}{l} S1W \succ_{p^1} C \\ S1W \succ_{p^1} L \\ L \sim_{p^1} C \end{array}$$

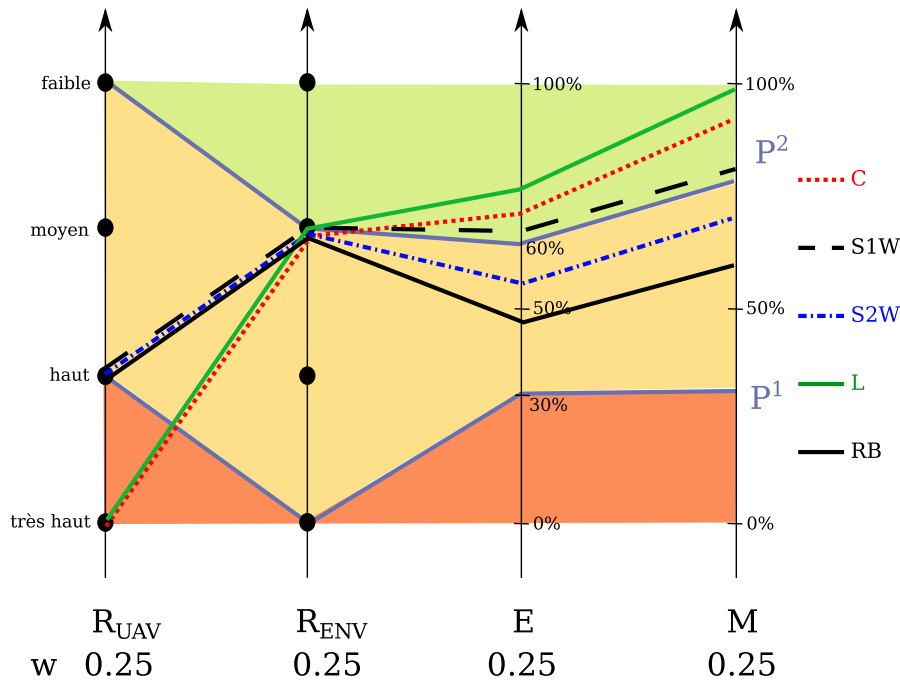


Figure 4.16 – Représentation des 5 actions dans le modèle SRMP de l'opérateur #2

Le classement final est ainsi $S1W \succ C \sim L \succ RB \sim S2W$, donc le drone choisit de sauter un waypoint puisqu'il estime que c'est trop risqué, ce qui respecte bien les priorités de l'opérateur qui tient aux résultats de la mission tout en restant prudent. Les deux actions qui suivent (C) et (L) aussi respectent ces priorités, mais elles sont plus risquées par rapport à la première.

4.6 Résumé et mise en perspective

Dans le but de valider notre modèle de décisions multi-niveaux qui prend en considération les préférences d'un opérateur avant de l'intégrer dans vrai drone, nous proposons dans ce chapitre un simulateur de drone autonome utilisé comme preuve de concept.

Le simulateur a pour but d'exécuter une mission prédéfinie d'un drone autonome en utilisant notre moteur de décisions multi-niveaux. Dans cette partie de notre travail, nous nous focalisons sur le niveau 2 du moteur décision où il y a un changement dans le plan de la mission et le drone doit choisir une action de haut niveau (continuer, atterrir, ...) parmi une liste d'actions possibles à exécuter. Dans le moteur de décisions proposé dans le chapitre 3 ce niveau est déclenché par le niveau 1, l'apparition d'un événement évalué comme perturbateur (obstacle, panne, ...). Cela est remplacé dans notre simulateur par un seul événement, l'arrivée à un waypoint. Donc à chaque waypoint le drone doit choisir une action de haut niveau.

Le simulateur est développé en MATLAB parce qu'il offre des boîtes à outils qui facilitent

l'implémentation des différentes fonctionnalités liées à la navigation et l'environnement des véhicules aérospatiaux. Le « Thales Watchkeeper WK450 » est utilisé comme modèle de drone dans le simulateur puisque c'est un drone autonome qui peut exécuter des missions de durée allant jusqu'à 17h. Mais le choix de ce drone nous a obligé à le modéliser comme un point qui est soumis aux mêmes contraintes physiques, vu que la tâche de la modélisation exacte de ce drone est trop complexe et sort du cadre de notre thèse.

La modélisation du drone représenté par un point est détaillée dans ce chapitre : la modélisation des forces qui agissent sur le drone, le mode de vol, le déplacement, la vitesse. Nous détaillons aussi la modélisation d'une mission où nous considérons que le niveau 2 est déclenché à l'arrivée à un waypoint. Dans ce cas l'ensemble des actions possibles sont évaluées sur de différents critères choisis (risque pour le drone et pour l'environnement, la consommation énergétique et la réalisation de la mission). Deux cartes géographiques de risque sont utilisées pour l'évaluation des actions sur les critères de risque. Un modèle de consommation d'énergie est développé pour estimer la consommation d'énergie de chaque action. Après le calcul de ces évaluations, un modèle SRMP pré-configuré selon les préférences d'un opérateur est intégré dans le simulateur et exécuté afin de choisir l'action à exécuter.

Nous présentons ensuite l'interface d'utilisateur graphique qui permet à l'opérateur de voir les détails de l'exécution de la mission et les informations liées au contexte du drone et son environnement ainsi que l'évaluation des différentes actions à chaque waypoint et l'historique des actions exécutées.

Nous finissons ce chapitre par un exemple d'exécution de mission de surveillance sur la ville de Brest en utilisant deux modèles SRMP différents qui représentent les préférences de deux opérateurs différents. Suite à cette simulation, nous remarquons bien que le drone réagit selon les préférences de chaque opérateur. Grâce à notre simulateur, il est possible de faire un audit de la mission, afin d'expliquer les décisions prises par le drone. Nous illustrons cela par quelques décisions, et soulignons les différences entre les deux exécutions, en fonction des profils des deux opérateurs.

À la fin de ce chapitre, nous pensons avoir mis en évidence l'intérêt de notre proposition. Grâce au simulateur, nous avons pu illustrer le fonctionnement de notre moteur de décisions. L'opérateur reste au centre de l'exécution, car toutes les décisions prises par le drone peuvent être expliquées. En cas de doutes, l'opérateur pourrait même proposer des mises à jour dans le paramétrage du modèle SRMP.

Le modèle SRMP utilisé dans ce simulateur est pré-configuré dans une phase d'élicitation des préférences d'un opérateur. Dans le chapitre 5 nous présentons des méthodes d'apprentissage de ces préférences se basant sur une base de paires d'apprentissage complète (apprentissage par lot). Dans le chapitre 6, une élicitation incrémentale du modèle SRMP est proposée. À chaque itération l'opérateur est confronté à une paire d'actions choisie par une heuristique de sélection. Il doit alors exprimer une préférence ou une indifférence sur la paire en question. Ceci doit notamment permettre de réduire son effort lors de la phase d'élicitation des préférences, comparée à une méthode qui utilise une grosse base d'apprentissage (telles que celles présentées au chapitre 6).

Troisième partie

Élicitation par lot et incrémentale des paramètres de modèles SRMP

Chapitre 5

Élicitation par lot des paramètres de modèles SRMP

Sommaire

5.1 État de l'art et étude d'expressivité	82
5.1.1 Programme linéaire mixte en nombres entiers	82
5.1.2 Expressivité des modèles SRMP	86
5.1.3 Problème de satisfaisabilité booléenne (SAT)	88
5.1.4 Matheuristique	88
5.2 Proposition d'un algorithme génétique	90
5.2.1 Généralités	90
5.2.2 Proposition des variantes d'un algorithme génétique	92
5.2.3 Étude expérimentale	96
5.2.4 Algorithme génétique d'apprentissage	105
5.2.5 Comportement de l'algorithme génétique face à de nouvelles données	105
5.3 Résumé et mise en perspective	106

Dans ce chapitre nous présentons les différents modèles d'inférence utilisés pour éliciter les paramètres d'un modèle SRMP dans le cas où les préférences du décideur sont exprimées sous la forme de préférences sur des paires d'alternatives, et où l'ensemble de ces paires est donné en entrée à l'algorithme (en un lot). Dans la littérature, ces modèles appartiennent à deux catégories : (1) les méthodes exactes, et (2) les méthodes approchées. Comme leur nom l'indique, la première catégorie de modèles permet de trouver une solution optimale au problème d'apprentissage des préférences. Nous identifions deux techniques dans la littérature : la programmation linéaire mixte en nombres entiers (MIP) et le problème de satisfaisabilité booléenne (SAT). Pour la deuxième catégorie, nous identifions dans la littérature une matheuristique qui est utilisée pour trouver une solution approchée en combinant la programmation mathématique avec une métaheuristique. Nous proposons ensuite une nouvelle approche métaheuristique, basée sur un algorithme génétique, qui permet d'éliciter les paramètres d'un modèle SRMP à partir d'un ensemble de comparaisons par paires. Nous montrons notamment que les paramètres de l'algorithme génétique influent peu sur la qualité de la solution obtenue, et qu'avec un nombre d'itérations peu élevé il est possible d'arriver à une bonne qualité de la solution.

5.1 État de l'art et étude d'expressivité

5.1.1 Programme linéaire mixte en nombres entiers

[Olteanu *et al.*, 2018] proposent une méthodologie pour inférer un modèle SRMP à partir d'une liste de comparaisons binaires donnée par le décideur. L'algorithme d'inférence utilise un programme linéaire mixte en nombres entiers (MIP) pour déterminer les paramètres d'un modèle SRMP :

- les k profils de référence $\mathcal{P} = \{p^1, \dots, p^k\}$;
- les poids des critères $w_j, j \in \mathcal{M}$;
- l'ordre lexicographique des profils σ .

Sans perte de généralités, nous supposons que les évaluations des alternatives sont dans $[0,1]$ et que le décideur préfère les valeurs élevées. Cela simplifie la présentation du MIP, mais il est évident que le MIP peut être étendu au cas où les échelles de valeurs sont hétérogènes, et où certains critères sont à maximiser et d'autres à minimiser.

La liste des comparaisons binaires fournie par le décideur est composée de deux sous-ensembles \mathcal{Pr} et \mathcal{In} où \mathcal{Pr} contient les paires d'alternatives (a, b) pour lesquels le décideur a exprimé une préférence de a par rapport à b , alors que \mathcal{In} inclut les paires d'indifférence c'est à dire l'ensemble des paires d'alternatives (a, b) telles que le décideur considère que a et b sont indifférents. \mathcal{A} est l'ensemble de toutes les alternatives.

Pour déterminer l'ordre lexicographique de k profils de références donnés, les auteurs examinent les $(k-1)!$ ordres lexicographiques possibles pour identifier le meilleur modèle SRMP qui respecte les préférences du décideur.

Pour un ordre lexicographique donné, déterminer si un modèle SRMP satisfait les préférences (\mathcal{Pr}) et les indifférences (\mathcal{In}) données par le décideur revient à résoudre un programme linéaire mixte en nombres entiers (MIP). Nous détaillons dans la suite ce programme, proposé par [Olteanu *et al.*, 2018], étant donné qu'il nous servira dans le chapitre 6 pour déterminer une heuristique de sélection de paires pour un processus d'élicitation incrémentale des préférences.

Pour modéliser qu'une alternative $a \in \mathcal{A}$ est au moins aussi bonne qu'un profil de référence $p^h, h = 1, \dots, k$ pour un critère donné $j \in \mathcal{M}$ nous utilisons une variable binaire $\delta_j(a, p^h)$ telle que :

$$\delta_j(a, p^h) = \begin{cases} 1 & , \text{si } a_j \geq p_j^h \\ 0 & , \text{sinon.} \end{cases} \quad (5.1)$$

Cette définition mène aux contraintes linéaires suivantes :

$$\begin{cases} \delta_j(a, p^h) \geq a_j - p_j^h + \gamma \\ a_j - p_j^h \geq \delta_j(a, p^h) - 1 \\ \delta_j(a, p^h) \in \{0, 1\} \end{cases} \quad (5.2)$$

où $\gamma > 0$ est une valeur très petite pour modéliser les inégalités. La première contrainte fixe $\delta_j(a, p^h)$ à 1 si $a_j \geq p_j^h$ et la deuxième sert à contraindre $\delta_j(a, p^h)$ à 0 si $a_j < p_j^h$.

Pour trouver les poids des critères pour $a \succsim p^h$, les auteurs définissent d'autres variables $\omega_j(a, p^h)$ de sorte que :

$$\omega_j(a, p^h) = \begin{cases} w_j & , \text{si } a_j \geq p_j^h \\ 0 & , \text{sinon.} \end{cases} \quad (5.3)$$

Cette définition mène aux contraintes linéaires suivantes :

$$\begin{cases} \omega_j(a, p^h) \geq 0 \\ \delta_j(a, p^h) \geq \omega_j(a, p^h) \\ w_j \geq \omega_j(a, p^h) \\ \omega_j(a, p^h) \geq \delta_j(a, p^h) + w_j - 1 \end{cases} \quad (5.4)$$

Les contraintes linéaires servent à donner une valeur minimum à $\omega_j(a, p^h)$ entre w_j et $\delta_j(a, p^h)$ et donc $\omega_j(a, p^h) = w_j$ si $a_j \geq p_j^h$, 0 sinon.

Ces contraintes et ces variables présentées dans les équations (5.1) et (5.3) pour un ordre lexicographique σ donné sont utilisées pour modéliser les préférences du décideur de la forme $a \succ b$ ($(a, b) \in \mathcal{Pr}$). Il s'agit de trouver un profil $p^{\sigma(i)}$, $i = 1, \dots, k$, de sorte que :

$$\begin{cases} \sum_{j \in \mathcal{M}} \omega_j(a, p^{\sigma(i)}) > \sum_{j \in \mathcal{M}} \omega_j(b, p^{\sigma(i)}) \\ \sum_{j \in \mathcal{M}} \omega_j(a, p^{\sigma(h)}) = \sum_{j \in \mathcal{M}} \omega_j(b, p^{\sigma(h)}), \forall h \in 1..i-1 \end{cases} \quad (5.5)$$

Un ensemble de $k-1$ variables $s(a, b, p^h)$, $\forall (a, b) \in \mathcal{Pr}, \forall p^h \in \mathcal{P}$ est utilisé pour modéliser l'affirmation préférentielle du décideur $a \succ b \forall (a, b) \in \mathcal{Pr}$. Si $p^{\sigma(1)}$ est le premier profil selon l'ordre lexicographique alors l'ensemble des contraintes suivant est défini :

$$\begin{cases} \sum_{j \in \mathcal{M}} \omega_j(a, p^{\sigma(1)}) \geq \sum_{j \in \mathcal{M}} \omega_j(b, p^{\sigma(1)}) + \gamma - s(a, b, p^{\sigma(1)}) \cdot (1 + \gamma) \\ \sum_{j \in \mathcal{M}} \omega_j(a, p^{\sigma(1)}) \geq \sum_{j \in \mathcal{M}} \omega_j(b, p^{\sigma(1)}) - (1 - s(a, b, p^{\sigma(1)})) \\ \sum_{j \in \mathcal{M}} \omega_j(a, p^{\sigma(1)}) < \sum_{j \in \mathcal{M}} \omega_j(b, p^{\sigma(1)}) + (1 - s(a, b, p^{\sigma(1)})) \end{cases} \quad (5.6)$$

La première contrainte est utilisée pour modéliser $\sum_{j \in \mathcal{M}} \omega_j(a, p^{\sigma(1)}) > \sum_{j \in \mathcal{M}} \omega_j(b, p^{\sigma(1)})$ et les deux autres pour $\sum_{j \in \mathcal{M}} \omega_j(a, p^{\sigma(1)}) = \sum_{j \in \mathcal{M}} \omega_j(b, p^{\sigma(1)})$. La variable $s(a, b, p^{\sigma(1)})$ agit comme un curseur entre les deux éléments, le premier est vrai pour $s(a, b, p^{\sigma(1)}) = 0$ et $s(a, b, p^{\sigma(1)}) = 1$ pour le second .

Ces contraintes peuvent être étendues pour les autres profils de référence en respectant l'ordre lexicographique. Elles doivent être relâchées pour tous les profils dans l'ordre qui suivent le profil

$(p^{\sigma(i)})$ capable de discriminer entre a et b . Ce qui permet de généraliser l'équation (5.6) pour $\forall h \in 2, \dots, k-1$:

$$\begin{cases} \sum_{j \in \mathcal{M}} \omega_j(a, p^{\sigma(h)}) \geq \sum_{j \in \mathcal{M}} \omega_j(b, p^{\sigma(h)}) + \gamma - s(a, b, p^{\sigma(h)}) \cdot (1 + \gamma) - (1 - s(a, b, p^{\sigma(h-1)})) \\ \sum_{j \in \mathcal{M}} \omega_j(a, p^{\sigma(h)}) \geq \sum_{j \in \mathcal{M}} \omega_j(b, p^{\sigma(h)}) - (1 - s(a, b, p^{\sigma(h)})) - (1 - s(a, b, p^{\sigma(h-1)})) \\ \sum_{j \in \mathcal{M}} \omega_j(a, p^{\sigma(h)}) < \sum_{j \in \mathcal{M}} \omega_j(b, p^{\sigma(h)}) + (1 - s(a, b, p^{\sigma(h)})) + (1 - s(a, b, p^{\sigma(h-1)})) \end{cases} \quad (5.7)$$

En ajoutant les variables $s(a, b, p^{\sigma(h-1)})$ à la fin des 3 contraintes, les contraintes du profil $p^{\sigma(h)}$ sont maintenues uniquement si le profil précédent selon l'ordre lexicographique $p^{\sigma(h-1)}$ n'a pas pu modéliser la relation de préférence entre a et b c'est à dire $s(a, b, p^{\sigma(h-1)}) = 1$. Par contre, si $p^{\sigma(h-1)}$ a pu modéliser la relation de préférence, $s(a, b, p^{\sigma(h-1)}) = 0$, alors les contraintes sur $p^{\sigma(h)}$ sont relâchées. Le dernier profil $p^{\sigma(k)}$ n'était pas inclus dans les équations précédentes, car il ne nécessite qu'une seule contrainte :

$$\sum_{j \in \mathcal{M}} \omega_j(a, p^{\sigma(k)}) \geq \sum_{j \in \mathcal{M}} \omega_j(b, p^{\sigma(k)}) + \gamma - s(a, b, p^{\sigma(k-1)}) \cdot (1 + \gamma) \quad (5.8)$$

Cette contrainte force le dernier profil de l'ordre lexicographique à modéliser $a \succ b$ lorsque tous les profils précédents n'ont pu le faire.

Pour une indifférence $a \sim b$, $(a, b) \in \mathcal{I}n$ une seule contrainte est utilisée pour chaque profil :

$$\sum_{j \in \mathcal{M}} \omega_j(a, p^{\sigma(h)}) = \sum_{j \in \mathcal{M}} \omega_j(b, p^{\sigma(h)}), \forall h \in 1, \dots, k \quad (5.9)$$

Le programme linéaire mixte en nombres entiers complet est présenté dans le tableau 5.1. Les contraintes (8) et (9) servent à fixer la somme des poids à 1 et à les rendre non nuls. Les trois contraintes qui suivent limitent également les évaluations des profils de référence à l'intervalle $[0, 1]$ et font respecter les contraintes de dominance entre eux. Les contraintes (13) et (14) correspondent à l'équation (5.2), tandis que les contraintes (15) à (18) correspondent à l'équation (5.4). Les cinq contraintes suivantes simplifient celles de l'équation (4), (5) et (6) en ajoutant deux variables fictives s^0 et s^k . s^0 est fixé à 1 de sorte que l'équation (5.6) soit modélisée, tandis que s^k est fixé à 0 pour modéliser l'équation (5.8). Enfin, la contrainte (24) est utilisée pour modéliser les paires d'alternatives indifférentes dans $\mathcal{I}n$.

Les auteurs de [Olteanu *et al.*, 2018] mènent également une étude de performance de l'algorithme d'apprentissage proposé. Ils étudient le temps de calcul, sa capacité à restaurer les comparaisons binaires données par le décideur, et son habilité à gérer des données bruitées.

Le temps de calcul augmente exponentiellement avec le nombre de comparaisons binaires. Il augmente également considérablement avec le nombre de profils. Cela complexifie l'utilisation de cette approche avec des données de cas réel (jusqu'à 100 comparaisons et 7 critères).

Dans les expériences présentées, les comparaisons binaires originales sont générées avec un modèle SRMP avec 10 profils alors que les modèles inférés utilisent un nombre de profils variant de 1 à 3. Donc pour la capacité du modèle inféré à restaurer l'ensemble d'apprentissages, ils

<i>Paramètres :</i>	
$\mathcal{A}, \mathcal{Pr}, \mathcal{In}, M, k, \sigma, \gamma$	(1)
$a_j \in [0, 1]$	$\forall a \in \mathcal{A}, \forall j \in M$ (2)
<i>Variables :</i>	
$w_j \in [0, 1]$	$\forall j \in M$ (3)
p_j^h	$\forall h \in 1..k, \forall j \in M$ (4)
$\delta_j(a, p^h) \in \{0, 1\}$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M$ (5)
$\omega_j(a, p^h) \in [0, 1]$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M$ (6)
$s(a, b, p^h) \in \{0, 1\}$	$\forall (a, b) \in \mathcal{Pr}, \forall h \in 0..k$ (7)
<i>Contraintes :</i>	
$s.t. \sum_{j \in M} w_j = 1$	(8)
$w_j \geq \gamma$	$\forall j \in M$ (9)
$p_j^1 \geq 0$	$\forall j \in M$ (10)
$1 \geq p_j^k$	$\forall j \in M$ (11)
$p_j^{h+1} \geq p_j^h$	$\forall h \in 1..k - 1, \forall j \in M$ (12)
$a_j - p_j^h \geq \delta_j(a, p^h) - 1$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M$ (13)
$\delta_j(a, p^h) \geq a_j - p_j^h + \gamma$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M$ (14)
$w_j \geq \omega_j(a, p^h)$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M$ (15)
$\omega_j(a, p^h) \geq 0$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M$ (16)
$\delta_j(a, p^h) \geq \omega_j(a, p^h)$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M$ (17)
$\omega_j(a, p^h) \geq \delta_j(a, p^h) + w_j - 1$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M$ (18)
$s(a, b, p^0) = 1$	$\forall (a, b) \in \mathcal{Pr}$ (19)
$s(a, b, p^k) = 0$	$\forall (a, b) \in \mathcal{Pr}$ (20)
$\sum_{j \in M} \omega_j(a, p^{\sigma(h)}) \geq \sum_{j \in M} \omega_j(b, p^{\sigma(h)}) + \gamma - s(a, b, p^{\sigma(h)}) \cdot (1 + \gamma) - (1 - s(a, b, p^{\sigma(h-1)}))$	$\forall (a, b) \in \mathcal{Pr}, \forall h \in 1..k$ (21)
$\sum_{j \in M} \omega_j(a, p^{\sigma(h)}) \geq \sum_{j \in M} \omega_j(b, p^{\sigma(h)}) - (1 - s(a, b, p^{\sigma(h)})) - (1 - s(a, b, p^{\sigma(h-1)}))$	$\forall (a, b) \in \mathcal{Pr}, \forall h \in 1..k$ (22)
$\sum_{j \in M} \omega_j(b, p^{\sigma(h)}) + (1 - s(a, b, p^{\sigma(h)})) + (1 - s(a, b, p^{\sigma(h-1)})) \geq \sum_{j \in M} \omega_j(a, p^{\sigma(h)})$	$\forall (a, b) \in \mathcal{Pr}, \forall h \in 1..k$ (23)
$\sum_{j \in M} \omega_j(a, p^{\sigma(h)}) = \sum_{j \in M} \omega_j(b, p^{\sigma(h)})$	$\forall (a, b) \in \mathcal{In}, \forall h \in 1..k$ (24)

Tableau 5.1 – MIP pour inférer un modèle SRMP

montrent que pour 7 critères et avec petit nombre de comparaisons binaires (≤ 40) un seul profil suffit. Diminuer le nombre de critères conduit à un modèle moins flexible et donc à un taux de restauration réduit. Par contre, augmenter le nombre de profils améliore la flexibilité du modèle et améliore donc le taux de restauration.

Les auteurs testent aussi la capacité de l'algorithme à trouver le modèle le plus proche du modèle original. Cela est traduit par le fait de retrouver le modèle qui reproduit le même ordre des données de test que le modèle original en utilisant le tau de *Kendall* [Kendall, 1938] τ qui mesure la corrélation de rang entre ces deux classements. τ mesure la corrélation de deux classements, et varie entre 1 et -1. Si les deux classements sont identiques, $\tau = 1$, alors que s'ils sont complètement inversés $\tau = -1$. Les résultats montrent que le τ augmente avec un plus grand nombre de comparaisons binaires. Si le nombre de critères augmente alors le modèle est plus flexible ce qui demande un plus grand nombre de comparaisons binaires pour atteindre un τ plus haut.

Dans le cas des données bruitées, le bruit est considéré comme un pourcentage d'erreur dans les comparaisons binaires. Ils testent différentes valeurs de pourcentage d'erreur : 5%, 10% et 15%. Les auteurs concluent que l'algorithme a un bon comportement face à des données bruitées.

Dans le chapitre 6 nous proposons une étude pour déterminer une heuristique de sélection de paires d'alternatives à proposer au décideur dans un processus d'élicitation incrémentale. Afin d'estimer le nombre de profils à utiliser dans l'élicitation, nous proposons ici d'étudier l'expressivité des modèles SRMP, et plus précisément l'intérêt d'utiliser un nombre élevé de profils.

5.1.2 Expressivité des modèles SRMP

L'expressivité d'un modèle SRMP est sa capacité à reproduire les préférences du décideur. Nous voulons dans cette section trouver le nombre de profils nécessaire pour garantir une bonne expressivité des modèles SRMP pour des problèmes avec au maximum 7 critères.

À cette fin, nous commençons par générer une série de 100 modèles SRMP avec $m \in \{3, 5, 7\}$ critères. Nous choisissons de générer ces modèles avec un grand nombre de profils de référence ($k = 10$), afin de simuler des modèles de préférence d'une grande complexité. Nous générons également de manière aléatoire 5000 alternatives avec leurs évaluations sur les critères. Pour terminer, nous appliquons les modèles SRMP à ces 5000 alternatives pour obtenir les ordres de ces 5000 alternatives selon les modèles considérés.

Nous utilisons alors le MIP présenté dans la section précédente pour apprendre les paramètres d'un modèle SRMP pour chaque ordre ainsi généré.

Nous illustrons dans le tableau 5.2 la moyenne et l'écart-type de la répartition du nombre de profils requis pour construire les relations entre ces alternatives. Nous rappelons ici que chaque paire d'alternatives est comparée aux profils de référence un par un, en nous arrêtant au premier profil dans l'ordre lexicographique qui place les deux alternatives dans une relation de préférence stricte ou d'indifférence.

Nous observons qu'un seul profil est capable de reproduire un grand nombre de comparaisons par paires pour les différents problèmes de 3, 5 ou 7 critères. Par exemple pour 3 critères, 67.15%

k	m					
	3		5		7	
	moyenne	écart-type	moyenne	écart-type	moyenne	écart-type
1	67.15	1.84	82.21	1.88	89.21	1.37
2	19.39	1.65	13.97	1.72	8.93	1.19
3	6.60	0.72	2.86	0.42	1.61	0.32
4	3.09	0.46	0.55	0.05	0.19	0.05
5	1.29	0.18	0.22	0.03	0.04	0.01
6	0.89	0.12	0.09	0.01	0.01	0.00
7	0.48	0.05	0.04	0.01	0.00	0.00
8	0.30	0.04	0.03	0.00	0.00	0.00
9	0.20	0.02	0.01	0.00	0.00	0.00
10	0.62	0.03	0.02	0.00	0.00	0.00

Tableau 5.2 – La moyenne et l'écart-type (en pourcentage) de la répartition des paires d'alternatives par rapport au nombre de profils nécessaires à leur construction .

en moyenne des comparaisons par paires qui peuvent être reconstruites à partir d'un seul profil avec un écart-type de 1.84. Cette valeur augmente à 89.21% (avec un écart type de 1.37) pour des problèmes de 7 critères.

Nous remarquons que plus le nombre de profils augmente plus le pourcentage de la moyenne diminue. Par exemple, pour des problèmes de 5 critères, 82.21% des comparaisons par paires nécessitent 1 seul profil, 13.97% en nécessitent 2, et seulement 2.86% en nécessitent 3. La proportion de comparaisons par paires nécessitant plus de 3 profils devient négligeable.

Les résultats montrent aussi que pour les différentes tailles de problèmes (3, 5 et 7 critères), l'inférence des modèles avec plus que 3 profils n'offre pas une amélioration significative en ce qui concerne l'expressivité des modèles SRMP. Plus de 93% des comparaisons par paires utilisent au plus 3 profils pour reconstruire leur relation de préférence.

Afin de tenir compte de tout biais causé par les alternatives dominées, nous présentons dans le tableau 5.3 la moyenne et l'écart-type en pourcentage de la répartition des paires d'alternatives dominées par rapport au nombre de profils nécessaires pour les discriminer entre eux. Dans notre étude les alternatives dominées représentent 24.95% en moyenne (pour les 100 modèles SRMP générés) de l'ensemble des 5000 alternatives pour les problèmes de 3 critères, 6.26% en moyenne des l'ensemble des 5000 alternatives pour les problèmes de 5 critères et 1.56% en moyenne des l'ensemble des 5000 alternatives pour les problèmes de 7 critères. Comme nous pouvons nous y attendre, pour les problèmes avec moins de critères, nous avons un pourcentage total plus élevé d'alternatives qui se dominent.

Nous observons que la plupart des dominances sont reconstruites à l'aide d'un seul profil et que le reste n'apporte pas beaucoup d'amélioration. Pour le pire cas où nous avons le plus de dominances, le cas des problèmes avec 3 critères, $67.09\% + 18.96\% + 7.33\% = 93.38\%$ des dominances existantes sont modélisables par un modèle SRMP à 3 profils.

Ces résultats nous permettent de conclure qu'en pratique, au maximum 3 profils de référence seront nécessaire pour garantir une bonne expressivité d'un modèle SRMP recherché, pour un problème avec au maximum 7 critères.

k	m					
	3		5		7	
	moyenne	écart-type	moyenne	écart-type	moyenne	écart-type
1	67.09	0.49	83.54	0.11	90.38	0.02
2	18.96	0.38	11.82	0.10	8.03	0.02
3	7.33	0.22	3.19	0.03	0.6	0.00
4	2.73	0.09	0.9	0.01	0.00	0.00
5	1.44	0.04	0.3	0.00	0.00	0.00
6	0.76	0.02	0.15	0.00	0.00	0.00
7	0.56	0.02	0.00	0.00	0.00	0.00
8	0.32	0.01	0.00	0.00	0.00	0.00
9	0.20	0.00	0.00	0.00	0.00	0.00
10	0.60	0.01	0.00	0.00	0.00	0.00

Tableau 5.3 – La moyenne et l'écart-type (en pourcentage) de la répartition des paires d'alternatives dominées par rapport au nombre de profils nécessaires à leur construction.

5.1.3 Problème de satisfaisabilité booléenne (SAT)

[Belahcene *et al.*, 2018] proposent une nouvelle procédure pour éliciter les paramètres d'un modèle RMP (Ranking with Multiple reference Profils) à partir d'un ensemble de comparaisons binaires \mathcal{L} en utilisant un problème de satisfaisabilité booléenne (SAT).

Les auteurs formulent les contraintes liées à la monotonie de l'échelle des critères, la dominance des profils, la relation de surclassement, l'ordre lexicographique et les préférences sous la forme d'un problème SAT pour apprendre les paramètres d'un modèle RMP. Ils limitent les préférences du décideur à des préférences strictes ($a \succ b, \forall (a, b) \in \mathcal{L}$).

Au moment de la rédaction de ce manuscrit, les premiers résultats présentés dans [Belahcene *et al.*, 2018] montrent que l'élicitation d'un modèle RMP avec des techniques de SAT est plus rapide qu'avec un MIP. Des travaux sont en cours pour valider ces résultats.

5.1.4 Matheuristique

L'utilisation des méthodes exactes pour éliciter les modèles SRMP nécessite un temps de calcul considérablement élevé. Ce temps est fortement lié à la taille des problèmes de décision (le nombre de critères) et à la complexité du modèle SRMP (le nombre de profils) ainsi que le nombre de comparaisons par paires. Cela limite l'utilisation de ces méthodes dans les applications réelles où les problèmes sont de grandes tailles, les modèles sont plus complexes et un grand nombre de comparaisons par paires est utilisé pour l'élicitation des préférences.

Pour de telles situations, des méthodes approchées peuvent être utilisées pour réduire le temps de calcul et traiter un plus grand nombre de comparaisons par paires.

[Liu *et al.*, 2014] proposent un matheuristique c'est-à-dire une approche combinant la programmation mathématique avec les principes d'une métaheuristique. Pour éliciter les paramètres d'un modèle SRMP les auteurs utilisent une métaheuristique pour trouver les profils et un programme

linéaire pour ajuster les poids des critères.

L'algorithme proposé commence par initialiser une population de solutions de manière aléatoire. Ensuite, à chaque itération, il fait évoluer ces solutions par des mutations, afin de créer une nouvelle population. Ces mutations se font selon différents opérateurs. Nous présentons les grandes lignes de cette proposition dans l'algorithme 1.

Algorithme 1 Matheuristique [Liu *et al.*, 2014]

Initialiser une population de N_{model} modèles SRMP

Évaluer chaque individu de la population

répéter

 Sélectionner les meilleurs individus de la population avec une probabilité μ

 Muter les individus sélectionnés avec une probabilité ξ

 Ajuster les poids des critères pour les individus mutés

 Évaluer les individus ajustés

 Substituer la population avec une probabilité ζ sous la contrainte de la taille de la population initiale.

jusqu'à au moins un des critères d'arrêt est vérifié

Dans l'algorithme 1 une partie des meilleurs modèles S-RMP de la population est sélectionnée au début de chaque itération. Ensuite, chaque modèle sélectionné (les parents) est soumis à l'opération de mutation qui permet de produire de nouveaux individus avec une certaine probabilité. Une fois les profils fixés, les poids des critères sont ajustés en utilisant un programme linéaire. Ceci crée les enfants des modèles parents. À la fin, la substitution consiste à sélectionner les meilleurs modèles parmi les parents et les enfants en fonction de leur évaluation. De nouveaux individus initialisés peuvent être intégrés à la nouvelle population si besoin.

Ils commencent l'élicitation par une initialisation d'une population de N_{model} modèles SRMP qui est fixée à 10 où le nombre de profils k est fixé à l'avance. Les profils sont générés aléatoirement, mais en respectant la dominance entre eux. L'ordre lexicographique est aussi généré aléatoirement. Concernant les poids des critères ils utilisent une approche proposée par [Butler *et al.*, 1997] de sorte que la somme des poids est égale à 1.

Un individu est évalué selon la précision de l'ordre qu'il a pu fournir par rapport à l'ordre initial donné par le décideur. Le modèle SRMP qui donne la précision la plus élevée sera sélectionné par une probabilité μ proportionnelle à son évaluation.

La mutation s'applique à tous les profils par ordre lexicographique. Pour chaque profil $p^{\sigma(h)}$, $h = 1, \dots, k$, chaque évaluation sur chaque critère $p_j^{\sigma(h)}$, $j \in \mathcal{M}$, une variation $\pm\theta_j^{\sigma(h)}$ est générée aléatoirement. Ensuite, ils évaluent l'impact, en termes de nombre de comparaisons binaires non respectées, causé par ce changement qui permet de calculer la probabilité de mutation ξ . Ce changement est appliqué au modèle courant selon cette probabilité ξ .

Un programme linéaire est proposé pour ajuster les poids des critères avec une fonction objective qui permet de maximiser le nombre de paires d'alternatives qui respectent les préférences du décideur qui sont données comme entrée.

La substitution permet de sélectionner les meilleurs individus avec le même opérateur de sélection déjà présenté avant. Une probabilité ζ est associée à chaque modèle et permet de déterminer s'il sera inclus dans la prochaine population ou non. S'il n'y a pas suffisamment d'individus qui sont assez bons pour être soumis, certains modèles SRMP nouvellement initialisés seront ajoutés à la nouvelle population sous la contrainte d'avoir la taille de population fixée initialement.

L'algorithme proposé a été mis à l'essai avec une grande quantité de données générées artificiellement qui simulent une variété de situations décisionnelles différentes. Il est capable de traiter jusqu'à 500 comparaisons par paires. De tels cas ne peuvent pas être résolus à l'aide d'un MIP. Le temps de calcul est proportionnel au nombre de critères impliqués et augmente avec le nombre de profils de référence. L'algorithme n'est pas capable d'apprendre un modèle S-RMP pour restaurer toutes les comparaisons par paires, mais il infère des modèles S-RMP qui restaurent près de 90% des préférences des données de test dans un temps de calcul beaucoup plus réduit par rapport au MIP.

5.2 Proposition d'un algorithme génétique

5.2.1 Généralités

Les algorithmes génétiques, proposés initialement par [Holland, 1975], sont des algorithmes d'optimisation s'appuyant sur des techniques dérivées de la génétique et des mécanismes d'évolution de la nature suivant le modèle darwinien : sélection, croisement, mutation.

Dans les algorithmes génétiques, un vocabulaire similaire à celui de la génétique naturelle est utilisé. Nous parlons ainsi d'individu dans une *population*. Un *individu* est représenté par un *chromosome* constitué de gènes qui contiennent les caractères héréditaires de l'individu. Les principes de *sélection*, *croisement*, *mutation* s'inspirent des processus naturels de même nom.

Pour un problème d'optimisation donné, un algorithme génétique recherche le ou les extrêmes d'une fonction définie sur un espace de données. Donc un individu dans ce cas représente un point de cette espace d'états et une solution potentielle. L'élaboration d'un algorithme génétique nécessite les cinq éléments suivants :

1. Un principe de codage des individus d'une population. Cette étape associe à chacun des points de l'espace d'état une structure de données. Elle se place généralement après une phase de modélisation mathématique du problème traité. Les codages binaires ont été très utilisés à l'origine. Les codages réels sont désormais largement utilisés, notamment dans les domaines applicatifs pour l'optimisation de problèmes à variables réelles.
2. Un mécanisme de génération de la population initiale. Ce mécanisme doit être capable de produire une population d'individus non homogène qui servira de base pour les générations futures. Le choix de la population initiale est important, car il peut rendre plus ou moins rapide la convergence vers un optimum. Dans le cas où l'on ne connaît rien du problème à résoudre, il est essentiel que la population initiale soit répartie sur tout le domaine de recherche.
3. Une fonction à optimiser. Celle-ci retourne une valeur appelée *fitness* ou *fonction d'évaluation* d'individus.

4. Des opérateurs permettant de diversifier la population au cours des générations et d'explorer l'espace de recherche. L'opérateur de croisement recompose les gènes d'individus existant dans la population et sert donc d'opérateur d'intensification, alors que l'opérateur de mutation a pour but de garantir une bonne exploration de l'espace de recherche
5. Des paramètres de dimensionnement : taille de la population, nombre total de générations/itérations ou critère d'arrêt, probabilités d'application des opérateurs de croisement et de mutation.

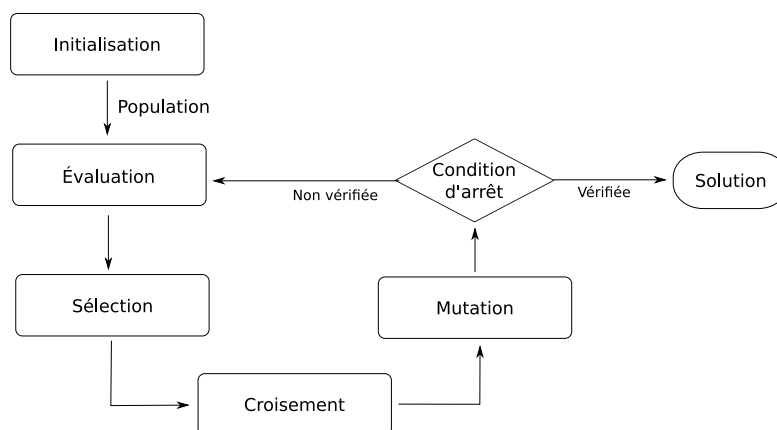


Figure 5.1 – Les étapes d'un algorithme génétique

L'algorithme génétique commence par une initialisation de la population comme le montre la figure 5.1 où les individus de la population sont générés. Pour passer d'une itération à une autre, les opérations suivantes sont effectuées. Dans un premier temps, les individus sont évalués en associant à chacun une valeur de la fonction d'évaluation. Cette valeur dépend de l'adaptation de l'individu au problème et sa pertinence. Ensuite, la population est reproduite par *sélection* où les bons individus se reproduisent mieux que les mauvais. Classiquement, cela est fait en sorte de conserver une population de taille constante. Un *croisement* est appliqué aux paires d'individus (les parents) d'une certaine proportion de la population pour en produire des nouveaux (les enfants). Un opérateur de *mutation* est également appliqué à une certaine proportion de la population. Enfin, les nouveaux individus sont évalués et intégrés à la population de l'itération suivante.

Plusieurs critères d'arrêt de l'algorithme sont possibles :

- atteindre un nombre d'itérations fixé a priori ;
- atteindre un temps d'exécution ;
- lorsque la population n'évolue plus suffisamment rapidement ;
-

Les différents points de l'algorithme génétique introduits ci-avant vont être détaillés pour notre cas d'élicitation par lot des modèles SRMP dans la section suivante.

5.2.2 Proposition des variantes d'un algorithme génétique

Notre algorithme génétique prend comme entrée une liste de comparaison par paires des alternatives \mathcal{L} (préférence stricte et indifférence) donnée par le décideur, le tableau de performances de l'ensemble des alternatives \mathcal{A} ($a \in \mathcal{A}, a \equiv (a_1, \dots, a_m)$) ainsi que le nombre de profils k . L'algorithme a besoin aussi du nombre d'éléments de la population nb_{pop} , la probabilité de mutation ξ et le nombre d'itérations nb_{itr} comme entrée.

Un individu, solution potentielle dans notre cas est un modèle SRMP défini par :

- k profils $P = \{p^1, \dots, p^k\}$ avec $p^h = \{p_j^h, \forall j \in \mathcal{M}\}$ qui respecte la dominance entre eux ;
- m poids de critères $w_j, \forall j \in \mathcal{M}$ où $\sum_{j \in \mathcal{M}} w_j = 1$;
- l'ordre lexicographique des profils σ qui est une permutation sur $1 \dots k$.

La première étape consiste à initialiser une population pop de nb_{pop} modèles SRMP, sachant que le nombre de profils k et le nombre de critères m sont donnés. Nous commençons par initialiser les k profils. Pour chaque critère j nous générons k nombres aléatoires $p_j^h, \forall j \in \mathcal{M}$ sur l'échelle d'évaluation du critère j , \mathcal{X}_j .

En fonction de la direction de préférence du critère j , ils sont ensuite classés par ordre croissant de sorte que :

$$p_j^1 \leq p_j^2 \leq \dots \leq p_j^k$$

ou par ordre décroissant de sorte que

$$p_j^k \leq p_j^{(k-1)} \leq \dots \leq p_j^1$$

afin de respecter la dominance entre eux. Pour k profils il y a $k!$ ordres lexicographiques possibles, nous choisissons un aléatoirement. L'initialisation des poids des critères se fait en générant m nombres aléatoires de sorte que leur somme est égale à 1 et ils sont strictement supérieurs à 0 et strictement inférieurs à 1 en utilisant la technique de [Butler *et al.*, 1997].

L'évaluation des « meilleurs » individus se fait en fonction de leur pertinence par rapport au problème. Dans le contexte d'apprentissage d'un modèle SRMP, cela correspond aux informations préférentielles données par le décideur. La fonction d'évaluation (ϕ) d'un individu (un modèle SRMP) correspond donc à la qualité de l'ordre donné par la solution. Elle est simplement définie comme le rapport entre le nombre de comparaisons binaires restaurées correctement par le modèle et le nombre total de comparaisons binaires fournies par le décideur :

$$\phi = \frac{\text{Nombre de comparaisons binaire restaurées correctement}}{\text{Nombre total de comparaisons binaires}}$$

La boucle de l'algorithme se répète jusqu'à avoir au moins une des solutions de la population qui est en mesure de restaurer correctement toutes les comparaisons binaires (c'est-à-dire $\phi = 1$) ou après un certain nombre d'itérations défini comme entrée nb_{itr} .

Dans notre algorithme l'étape de sélection n'a lieu qu'à partir la deuxième itération où un croisement et une mutation sont déjà effectués.

Nous proposons dans la suite de cette section différentes variantes de l'algorithme génétique, en proposant différents opérateurs de sélection, de croisement et de mutation. Les performances de toutes ces variantes seront détaillées dans la section 5.2.3 afin de proposer un choix motivé pour l'algorithme génétique.

Pour l'opérateur de sélection, ces variantes sont les suivantes :

- Sélection avec **élitisme total** : après avoir évalué chaque solution, nous les ordonnons selon la valeur de leur fonction d'évaluation ϕ . Cet opérateur permet de sélectionner les nb_{pop} premières (meilleures) solutions.
- Sélection avec **probabilité** : nous calculons pour chaque solution d'indice $i, i = 1, \dots, nb_{pop}$ la valeur de sa fonction d'évaluation ϕ_i et ensuite nous calculons sa probabilité μ_i de sélection qui dépend de ϕ_i ainsi que ϕ_{min} et ϕ_{max} qui sont la valeur minimale et maximale et la fonction d'évaluation de toutes les solutions.

$$\mu_i = \frac{\phi_i - \phi_{min}}{\phi_{max} - \phi_{min}}, \text{ avec}$$

$$\phi_{min} = \{\min \phi_i, \forall i \in 1, \dots, nb_{pop}\} \quad \phi_{max} = \{\max \phi_i, \forall i \in 1, \dots, nb_{pop}\}$$

- Sélection avec **élévation au carré** : un nombre aléatoire entre 0 et 1 est généré et comparé au carré d'évaluation d'une solution. Si ce nombre est plus petit alors cette solution est sélectionnée. Cette opération se répète jusqu'à arriver à atteindre la taille de la population nb_{pop} .
- Sélection à **la roulette** : Chaque solution occupe un secteur d'une roulette dont l'angle est proportionnel à son évaluation. Nous notons ϕ_i l'évaluation de la solution d'indice $i, i = 1, \dots, nb_{pop}$ et

$$\Phi = \sum_{i=1}^{nb_{pop}} \phi_i$$

la somme des valeurs de l'évaluation de l'ensemble des solutions.

$$Pi = \sum_{j=1}^i \phi_j$$

est une valeur affectée à chaque solution d'indice i . Un nombre aléatoire n est généré entre 0 et Φ . Nous sélectionnons la solution i sélectionnée qui a l'évaluation ϕ_i la plus proche de n . Cette opération se répète jusqu'à sélectionner nb_{pop} solutions.

L'étape de croisement consiste au début à choisir aléatoirement deux solutions parents (modèles SRMP) S_1 et S_2 pour générer des solutions enfants. Pour le croisement des profils nous notons les profils de chaque solution $P_{S1} = \{p_{S1}^1, \dots, p_{S1}^k\}$ et $P_{S2} = \{p_{S2}^1, \dots, p_{S2}^k\}$ et les poids des critères par w_j^{S1} et $w_j^{S2}, \forall j \in \mathcal{M}$.

Nous proposons deux façons de croiser des profils :

- **Croisement à un point** : nous choisissons un point de coupure c qui est un nombre aléatoire entre 1 et m . Un vecteur de permutation des indices de critères est aussi généré aléatoirement noté V . L'algorithme 2 permet de générer les deux profils enfants P_{E1} et P_{E2} .

Algorithme 2 Croisement à un point

```

pour  $h = 1$  à  $k$  faire
  pour  $j = 1$  à  $m$  faire
     $d := V(j)$ 
    si  $j \leq c$  alors
       $P_{E1,d}^h := P_{S1,d}^h$ 
       $P_{E2,d}^h := P_{S2,d}^h$ 
    sinon
       $P_{E1,d}^h := P_{S2,d}^h$ 
       $P_{E2,d}^h := P_{S1,d}^h$ 
    fin si
  fin pour
fin pour
  
```

- **Croisement à deux points** : c'est le même principe que le croisement précédent sauf que nous fixons deux points de coupure c_1 et c_2 avec $c_1 \leq c_2$. Pour générer les deux enfants, l'algorithme 3 est utilisé.

Algorithme 3 Croisement à deux points

```

pour  $h = 1$  à  $k$  faire
  pour  $j = 1$  à  $m$  faire
     $d := V(j)$ 
    si  $j \leq c_1$  et  $j \geq c_2$  alors
       $P_{E1,d}^h := P_{S1,d}^h$ 
       $P_{E2,d}^h := P_{S2,d}^h$ 
    sinon
       $P_{E1,d}^h := P_{S2,d}^h$ 
       $P_{E2,d}^h := P_{S1,d}^h$ 
    fin si
  fin pour
fin pour
  
```

Suite au croisement des deux profils parents, deux profils enfants sont générés. Ces deux derniers sont utilisés pour générer des solutions en combinant chacun des profils enfants avec les poids des critères de chaque solution parents et les profils de S_1 avec les poids de S_2 et les profils de S_2 avec les poids de S_1 . Deux situations se présentent alors :

- Si les ordres lexicographiques de S_1 et de S_2 sont identiques, cette procédure génère 6 solutions enfants,
- Sinon, cette procédure génère 12 solutions enfants en utilisant la même combinaison pour générer 6 solutions avec l'ordre lexicographique de S_1 et 6 de plus avec l'ordre lexicogra-

phique de S_2 .

Pour l'étape de mutation, nous avons différents opérateurs pour muter les profils, les critères et l'ordre lexicographique. La mutation des profils peut se faire de deux façons différentes :

- mutation **aléatoire** : Pour chaque $p_j^{\sigma(h)}$, $h = 1, \dots, k \forall j \in \mathcal{M}$, avec σ est l'ordre lexicographique des profils, nous générons un nombre aléatoire nb . Si nb est supérieur à la probabilité de mutation donnée comme entrée ξ alors pas de mutation et nous gardons la même valeur de $p_j^{\sigma(h)}$. Sinon, $p_j^{\sigma(h)}$ est muté en le remplaçant par un nombre aléatoire généré p de sorte qu'il respecte la dominance entre les profils. Si le critère j est à maximiser alors :

$$\begin{cases} p^{\sigma(h-1)} \leq p \leq p_j^{\sigma(h+1)} & , \text{ si } 1 < h < k \\ p \leq p_j^{\sigma(2)} & , \text{ si } h = 1 \\ p \geq p_j^{\sigma(k)} & , \text{ si } h = k \end{cases}$$

Si le critère j est à minimiser alors :

$$\begin{cases} p^{\sigma(h+1)} \leq p \leq p_j^{\sigma(h-1)} & , \text{ si } 1 < h < k \\ p \geq p_j^{\sigma(2)} & , \text{ si } h = 1 \\ p \leq p_j^{\sigma(k)} & , \text{ si } h = k \end{cases}$$

- mutation sur base des **impacts positifs** : nous utilisons ici la notion d'impact positif I^+ et impact négatif I^- afin de décider de muter $p_j^{\sigma(h)}$ ou non. I^+ est le nombre d'alternatives qui respectent leur les préférences du décideur par rapport au $p_j^{\sigma(h)}$ alors que I^- est le nombre d'alternatives qui ne les respectent pas.

Pour rappel, \mathcal{Pr} est l'ensemble des paires d'alternatives pour lesquelles le décideur a émis une préférence de l'une par rapport à l'autre.

Pour I^+ nous comptons le nombre d'alternatives a et b avec $(a, b) \in \mathcal{Pr}$ où :

$$a_j \geq p_j^{\sigma(h)} \text{ et } b_j \leq p_j^{\sigma(h)}.$$

Pour I^- nous comptons le nombre d'alternatives a et b avec $(a, b) \in \mathcal{Pr}$ où

$$a_j < p_j^{\sigma(h)} \text{ et } b_j > p_j^{\sigma(h)}.$$

Ces deux mesures sont utilisées pour calculer la probabilité de mutation :

$$\xi_I = \frac{I^+}{I^+ + I^-}.$$

Un nombre aléatoire est généré, s'il est plus grand que ξ_I alors il n'y a pas de mutation, sinon $p_j^{\sigma(h)}$ est remplacé par un nombre aléatoire généré de sorte qu'il respecte la dominance entre les profils.

Pour muter les poids des critères, nous générons un nombre aléatoire nb et le comparons à la probabilité de mutation. S'il est plus grand que ξ , alors aucune mutation n'est appliquée. Sinon, les critères peuvent être mutés de manières différentes :

- mutation **aléatoire** : elle permet de choisir deux critères au hasard et de permuter leur poids.
- mutation avec le **poids du critère suivant** : nous prenons le poids du critère i , w_i et de son successeur $i + 1$, w_{i+1} . Nous générons un nombre aléatoire w entre $-w_i$ et w_{i+1} . w_i est alors remplacé par $w_i + w$ et w_{i+1} par $w_{i+1} - w$. Ce processus s'applique successivement pour tout $i = 1, \dots, m - 1$.

La mutation de l'ordre lexicographique commence par choisir un élément de l'ordre au hasard, et le permute avec l'un de ses voisins.

5.2.3 Étude expérimentale

Pour trouver la bonne configuration de notre algorithme génétique présenté dans la section précédente, nous proposons de tester différentes combinaisons des opérateurs de chaque étape de l'algorithme.

5.2.3.1 Configurations étudiées

Comme déjà proposés dans la section précédente, nous avons un mécanisme d'initialisation d'une population qui permet la génération aléatoire de n_{pop} modèles SRMP. Dans notre problème d'optimisation le but est maximiser la fonction d'évaluation ϕ de la solution (modèle SRMP) à trouver.

Nous choisissons de tester quatre différentes configurations de l'algorithme génétique. Ces configurations permettent de tester des combinaisons des opérateurs de sélection, croisement et mutation proposés ci-avant. Nous fixons l'opérateur de croisement à celui de deux points et l'opérateur de mutation à aléatoire et nous varions l'opérateur de sélection.

Le tableau 5.4 présente les différents opérateurs utilisés pour chacune des quatre configurations.

Configuration	Op. Sélection	Op. croisement	Op. Mutation
#1	Aléatoire	Deux points	Aléatoire
#2	Élévation au carré	Deux points	Aléatoire
#3	Roulette	Deux points	Aléatoire
#4	Élitisme	Deux points	Aléatoire

Tableau 5.4 – Différentes configurations de l'algorithme génétique.

Pour chacune des configurations, deux paramètres peuvent être testés avec différentes valeurs :

- La taille de la population : $n_{pop} = 20, 60, 100, 140$;
- La probabilité de mutation : $P_{mut} = 0.1, 0.3, 0.5, 0.7, 0.9$.

5.2.3.2 Expériences

Pour une expérience d'une configuration de l'algorithme, une base de données de 100 paires d'alternatives est générée aléatoirement. Le décideur est remplacé ici par un modèle SRMP généré aussi aléatoirement M_{DM} , qui est utilisé pour comparer les paires d'alternatives (préférence stricte et indifférence), ce qui génère une liste de comparaison par paires des alternatives \mathcal{L} . Cette liste est utilisée comme entrée pour l'algorithme génétique pour construire un modèle SRMP M .

Nous exécutons cette expérience pour 100 bases de données artificielles différentes, composées chacune de 100 paires d'alternatives, évaluées sur différents nombres de critères ($m = 3, 5, 7$). Le choix de s'arrêter à 7 critères est lié au fait que dans des cas réels, un décideur humain va avoir beaucoup de difficultés à comparer des alternatives en prenant en compte simultanément au-delà de 7 critères. Différentes complexités (nombre de profils k) du modèle SRMP sont testées avec $k = 2, 3$. Ce choix est justifié par l'étude d'expressivité des modèles SRMP présentée dans la section 5.1.2 et qui montre que 3 profils suffisent en général pour représenter les préférences d'un décideur à l'aide d'un modèle SRMP tout en garantissant un bon niveau d'expressivité.

Ces expériences sont exécutées pour chacune des configurations d'algorithme génétique proposées dans le tableau 5.4. Dans la suite de ce travail, nous notons $kPmC$ les instances de problèmes générés avec k profils et m critères.

5.2.3.3 Résultats

La figure 5.2 montre pour des problèmes à 3 critères, générés avec 2 profils ($2P3C$), la moyenne sur 100 expériences de la valeur de la fonction d'évaluation ϕ à chaque itération pour les différentes tailles de la population ($n_{pop} = 20, 60, 100, 140$) et les différentes valeurs de la probabilité de mutation ($P_{mut} = 0.1, 0.3, 0.5, 0.7, 0.9$). Ces expériences sont réalisées avec la configuration #1 présentée dans le tableau 5.4. Nous remarquons dans cette figure que la valeur de ϕ converge à partir de 30 itérations environs et que l'amélioration de ϕ n'est plus significative au-delà d'une soixantaine d'itérations.

La figure 5.3 montre le même phénomène de convergence de la valeur moyenne de ϕ dans l'intervalle de 30 à 60 itérations pour les différentes tailles de population et valeurs de probabilité de mutation pour des problèmes à 7 critères, générés avec 2 profils ($2P7C$). Ces résultats sont obtenus en exécutant l'algorithme génétique avec la configuration #4.

Cette convergence se retrouve dans toutes les expériences que nous avons menées, avec toutes les configurations du tableau 5.4, les différentes tailles de populations et valeurs de la probabilité de mutation, et pour toutes les complexités de modèles (données par le nombre de profils et le nombre de critères). La fonction d'évaluation a l'air de converger au bout d'une trentaine d'itérations, comme dans les deux exemples présentés ci-avant. Les résultats détaillés sont produits dans l'annexe A.

Ceci nous mène à la première conclusion de notre étude. Le nombre d'itérations de notre algorithme génétique peut être fixé dans un intervalle de 30 à 60 d'itérations, indépendamment de la configuration et de la valeur des paramètres (taille de la population, probabilité de mutation).

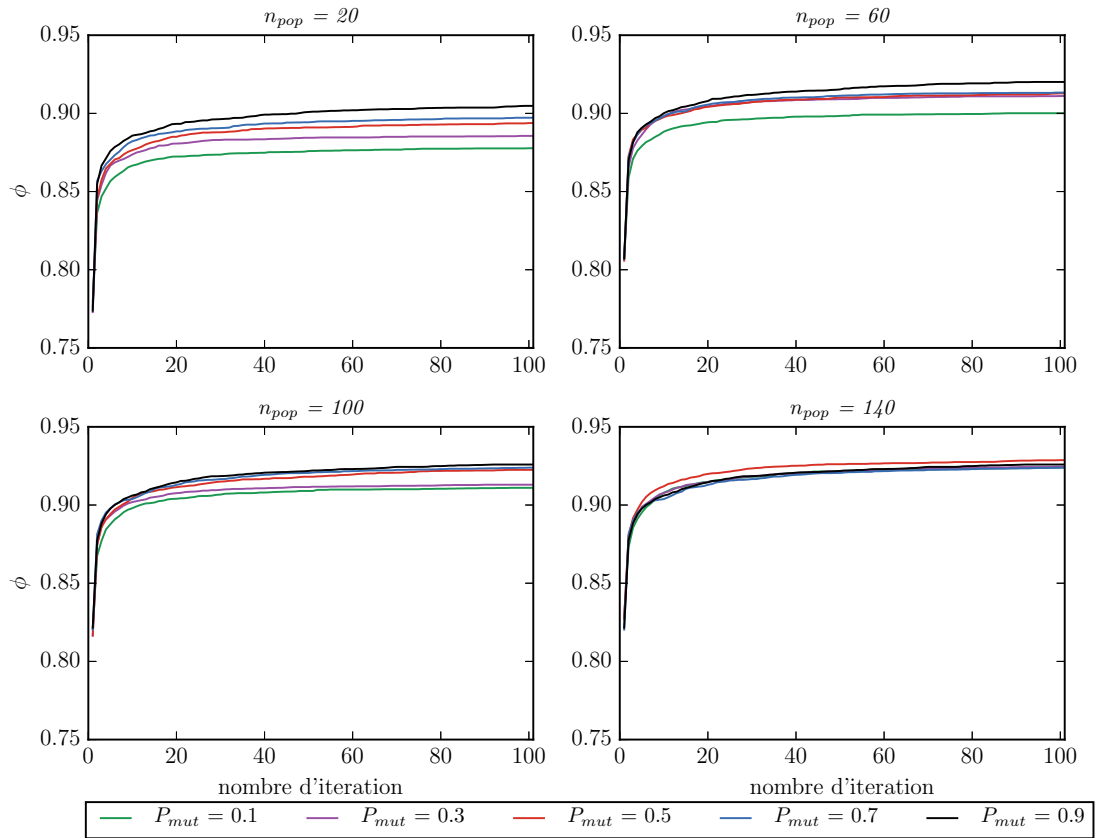
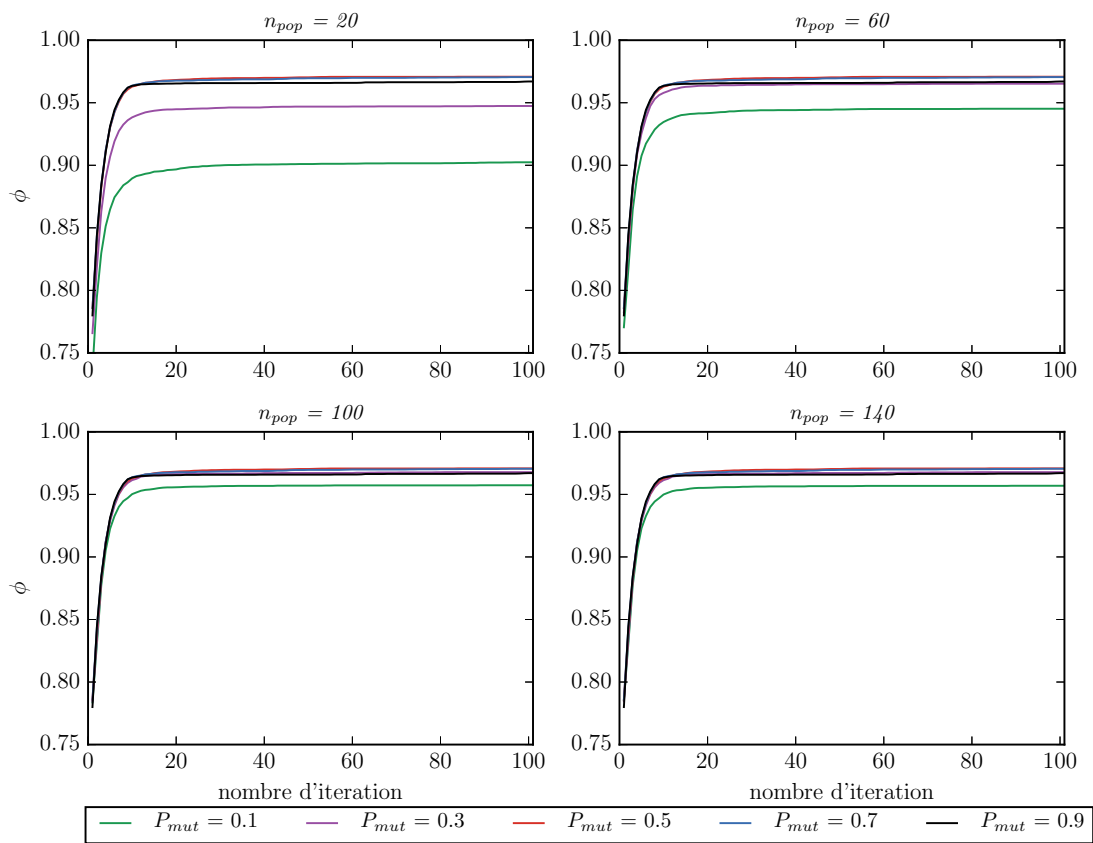


Figure 5.2 – Évaluation (ϕ) moyenne pour les problèmes $3P5C$ avec la configuration #1

Figure 5.3 – Évaluation (ϕ) moyenne pour les problèmes $2P7C$ avec la configuration #4

La figure 5.4 montre la valeur moyenne sur 100 expériences de la valeur de la fonction d'évaluation ϕ pour les quatre tailles de population différentes (20, 60, 100 et 140) à chaque itération. Ce sont les résultats pour les problèmes $2P5C$ avec la configuration #3 et une probabilité de mutation $P_{mut} = 0.3$. Nous remarquons que plus la taille de la population augmente, plus la valeur de ϕ augmente.

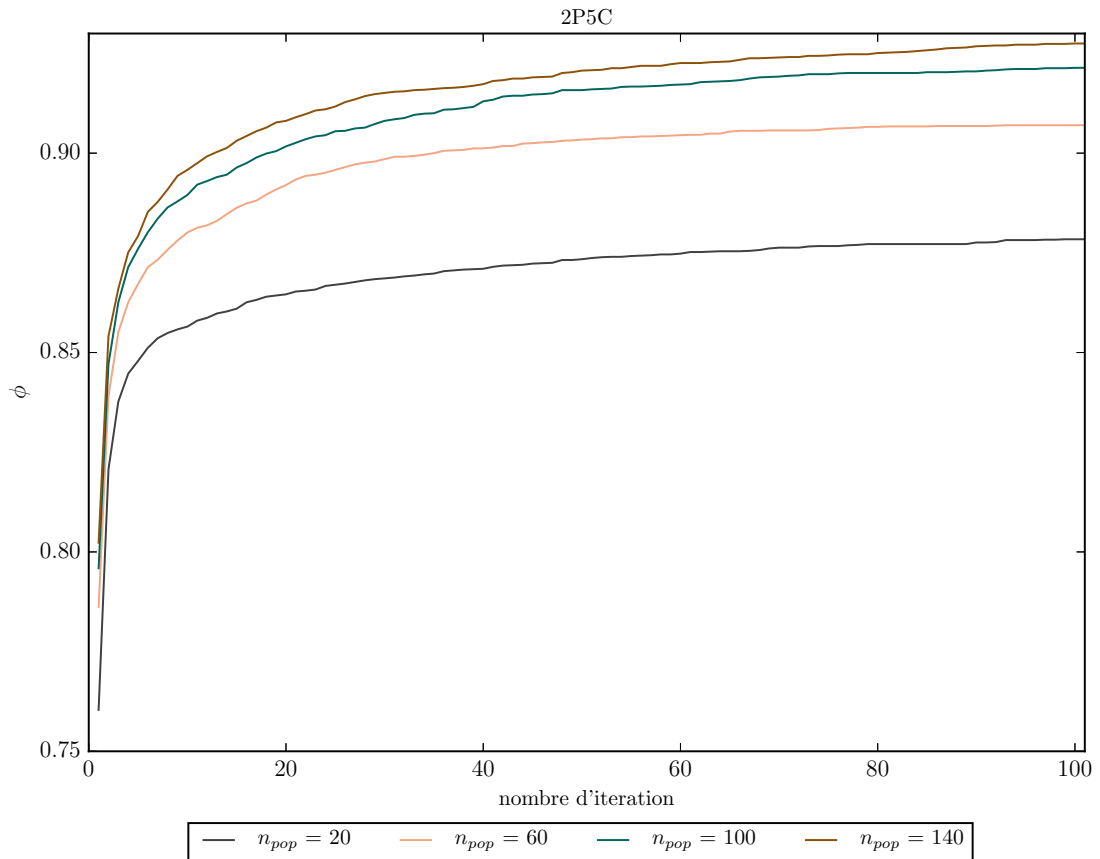


Figure 5.4 – Évaluation (ϕ) moyenne pour les problèmes $2P5C$ avec la configuration #3 et $P_{mut} = 0.3$

La figure 5.5 qui illustre la moyenne de ϕ pour les problèmes $3P3C$ avec la configuration #1 et une probabilité de mutation $P_{mut} = 0.7$ confirme bien l'observation précédente où la valeur de ϕ augmente proportionnellement avec la taille de la population.

Cette observation se vérifie pour toutes les expériences réalisées. Cela est illustré dans les figures présentées dans l'annexe A.

La deuxième conclusion de cette étude est donc qu'il faut choisir une population de grande taille pour avoir un modèle SRMP avec une qualité élevée.

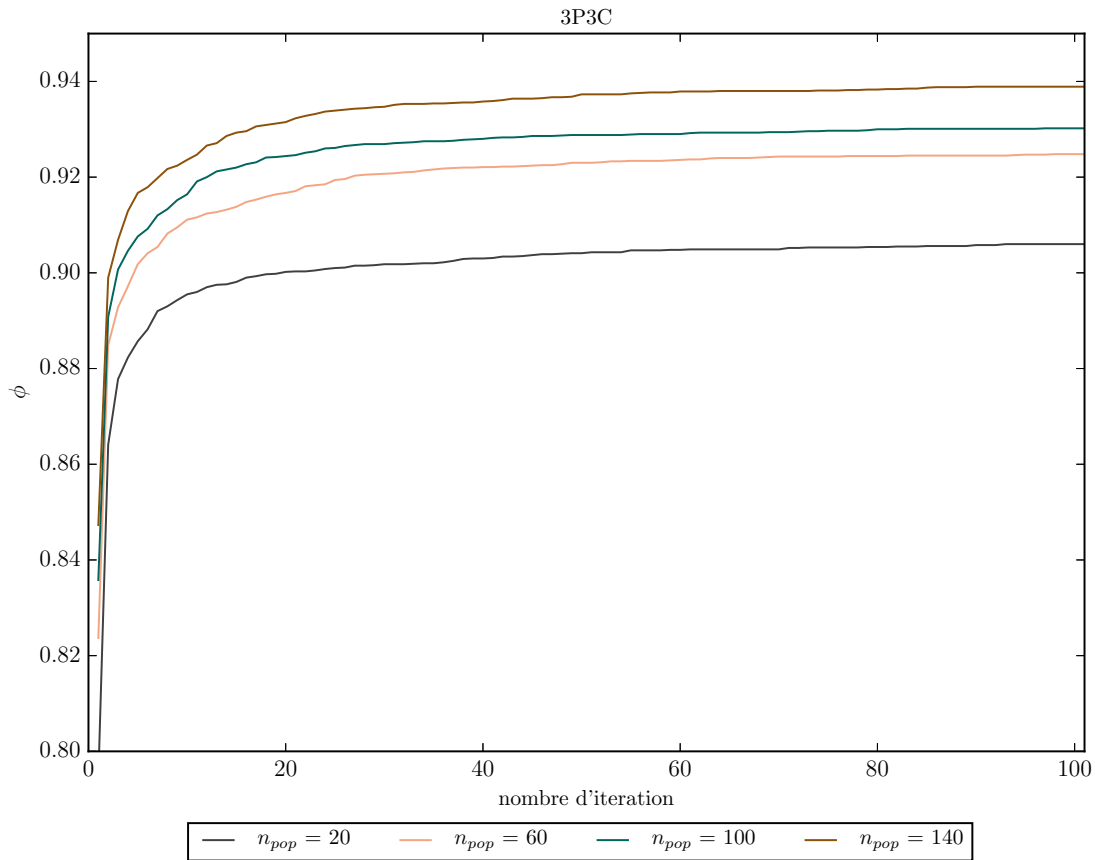
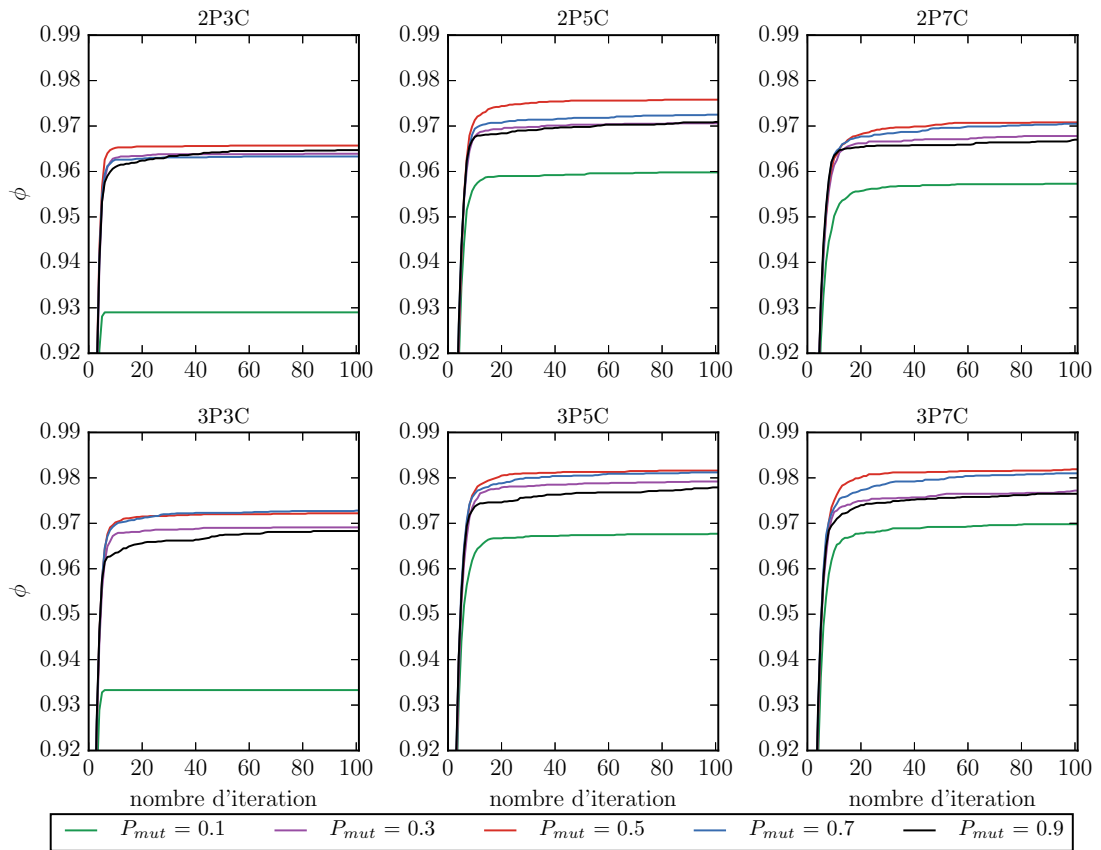


Figure 5.5 – Évaluation (ϕ) moyenne pour les problèmes $3P3C$ avec la configuration #1 et $P_{mut} = 0.7$

La figure 5.6 montre la moyenne de ϕ pour chaque itération pour différents problèmes ($m = 3, 5, 7$ et $k = 2, 3$) avec une taille de population fixée à 100 ($n_{pop} = 100$) et avec la configuration #4. Nous observons que la courbe de ϕ pour une probabilité de mutation égale à 0.1 (en vert) est toujours dominée par les autres courbes. Pour les autres probabilités il est difficile de distinguer la meilleure de manière absolue.

Figure 5.6 – Évaluation (ϕ) moyenne avec la configuration #4 et $n_{pop} = 100$

Nous observons le même phénomène pour la figure 5.7 qui illustre la moyenne de ϕ à chaque itération pour les différents problèmes avec 100 comme taille de population et avec la configuration #1. Nous observons à nouveau que la courbe de $P_{mut} = 0.1$ est dominée par les autres, mais qu'aucune autre courbe ne domine les autres pour toutes les complexités de problèmes.

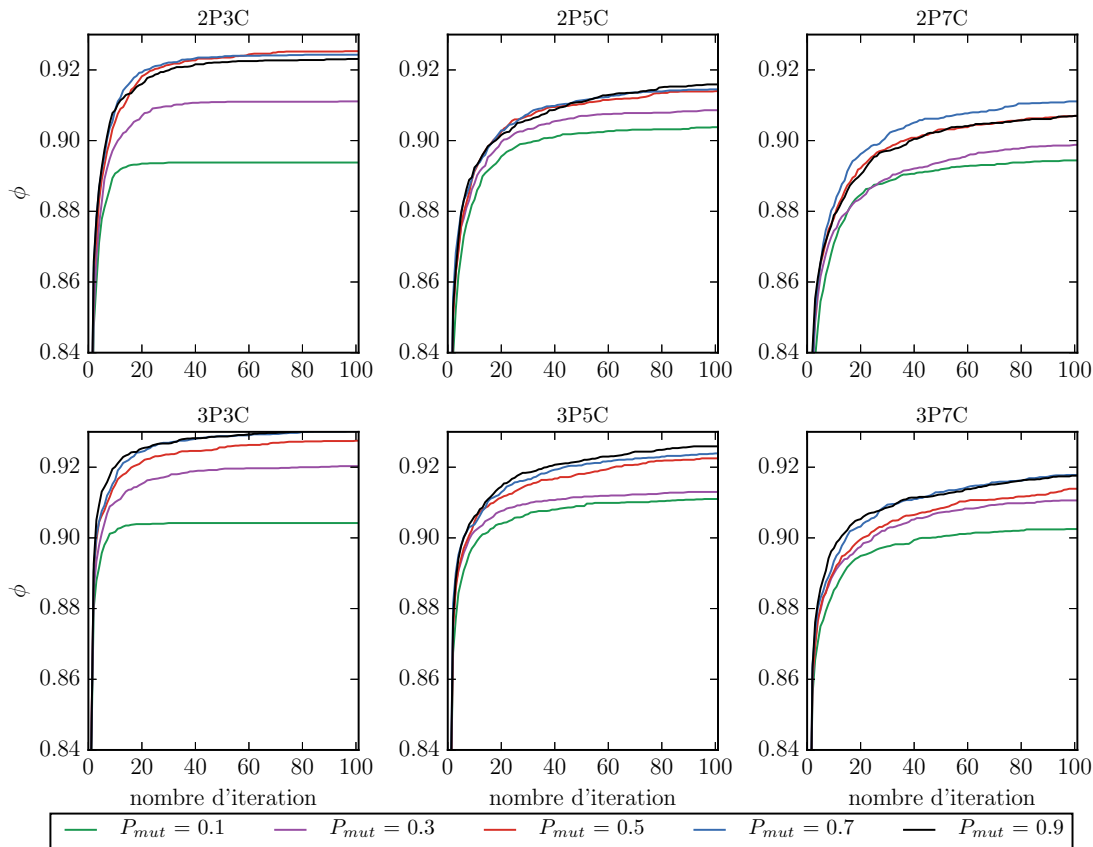


Figure 5.7 – Évaluation (ϕ) moyenne avec la configuration #1 et $n_{pop} = 100$

Nous en déduisons la troisième conclusion de notre étude : la probabilité de mutation doit être strictement supérieure à 0.3 , et plutôt tendre vers des valeurs élevées.

En tenant compte des observations obtenues jusqu'à ici, nous nous intéressons plus particulièrement aux résultats obtenus avec une taille de population égale à 100 et une probabilité de mutation égale à 0.7 afin d'essayer d'identifier la meilleure configuration parmi les quatre proposées dans le tableau 5.4.

La figure 5.8 montre l'évaluation (ϕ) moyenne pour les différentes configurations avec $n_{pop} = 100$ et $P_{mut} = 0.7$. Nous remarquons que la courbe de la configuration #4 où l'opérateur de sélection est l'élitisme totale domine toutes les autres courbes pour toute complexité de modèle (nombre de profils) et toute taille de problème (nombre de critères). Cela nous mène à choisir la configuration #4 pour notre algorithme génétique.

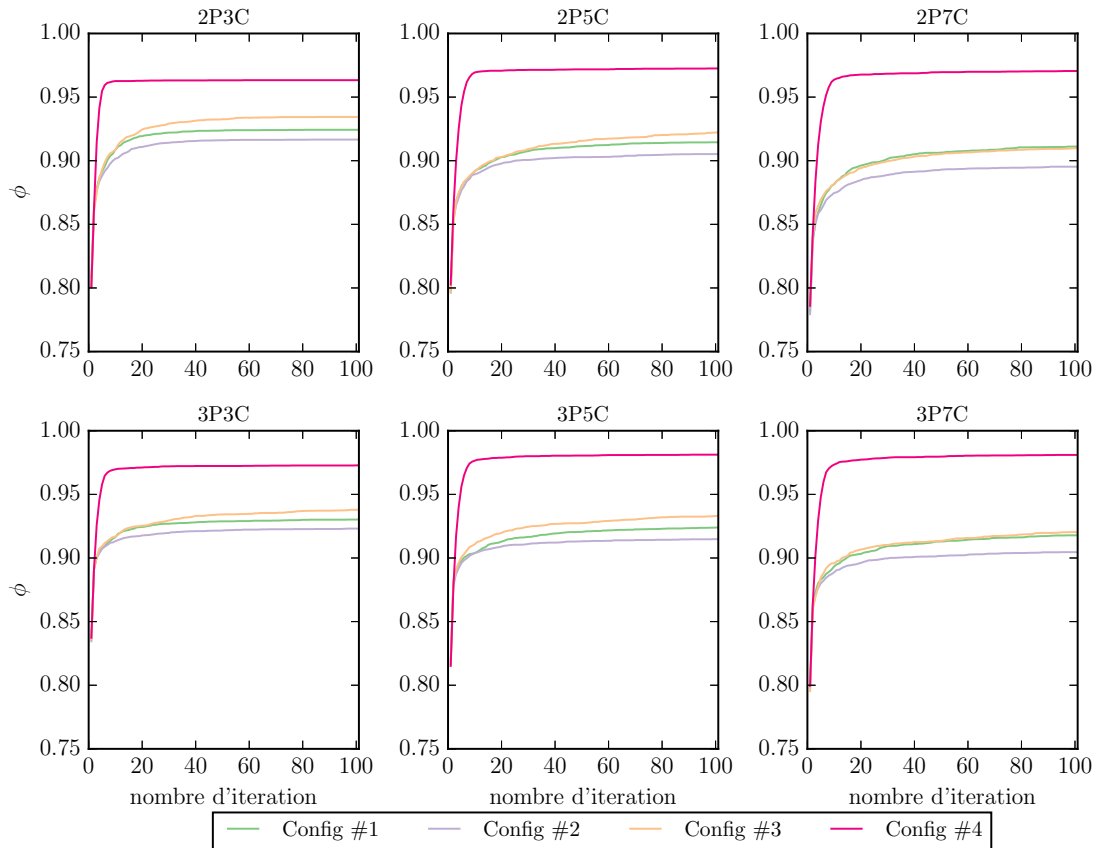


Figure 5.8 – Évaluation (ϕ) moyenne pour les différentes configurations avec $n_{pop} = 100$ et $P_{mut} = 0.7$

5.2.4 Algorithme génétique d'apprentissage par lot des préférences d'un modèle SRMP

À partir des observations obtenues des résultats des différentes expériences, nous proposons un algorithme génétique (algorithme 4) pour l'élicitation par lot des modèles SRMP.

Algorithme 4 Algorithme génétique pour l'élicitation par lot des modèles SRMP

Initialisation d'une population de $n_{pop} := 100$ modèles SRMP

Évaluation de chaque individu de la population

$nb_{itr} := 0$

$P_{mut} := 0.7$

$Model := \emptyset$

répéter

 Mutation des critères des individus sélectionnés avec une probabilité P_{mut}

 Mutation des profils des individus sélectionnés avec une probabilité P_{mut}

 Mutation de l'ordre lexicographique des individus sélectionnés avec une probabilité P_{mut}

 Croisement des individus mutés

 Sélection les meilleurs individus de la population

$Model :=$ meilleure individu

$nb_{itr} := nb_{itr} + 1$

jusqu'à $nb_{itr} = 100$ ou $\phi(Model)=1$

retourne $Model$

5.2.5 Comportement de l'algorithme génétique face à de nouvelles données

Comme pour les expériences précédentes (section 5.2.3.2) une base de données de 100 paires est générée aléatoirement et le décideur est remplacé par un modèle SRMP généré aléatoirement M_{DM} . Ces données sont utilisées comme entrée pour notre algorithme génétique (algorithme 4) pour éliciter un modèle SRMP M .

Afin d'évaluer le comportement du modèle SRMP élicité M par notre algorithme génétique (algorithme 4) face à de nouvelles alternatives non rencontrées pendant la phase d'élicitation une base D_{test} de 5000 alternatives de test est utilisée. Les alternatives sont classées à la fois par les deux modèles SRMP M_{DM} et M . Nous calculons ensuite le tau de *Kendall* τ qui mesure la corrélation de rang entre ces deux classements et qui varie entre 1 et -1. Ceci permet d'évaluer la capacité de M à ordonner correctement de nouvelles alternatives. Plus τ sera proche de 1, plus M réagira correctement face à ces nouvelles alternatives.

Cette expérience est exécutée pour 100 bases de données pour chacune des six configurations de problèmes possibles en faisant varier le nombre de critères $m = 3, 5, 7$ et le nombre de profils $k = 2, 3$.

La figure 5.9 représente la valeur moyenne et l'écart type du tau de *Kendall* sur ces 100 expériences pour les différents problèmes. Nous observons que les valeurs du tau de *Kendall* pour les différents problèmes sont proches de 1. Cela montre que notre algorithme se comporte bien

face à de nouvelles alternatives.

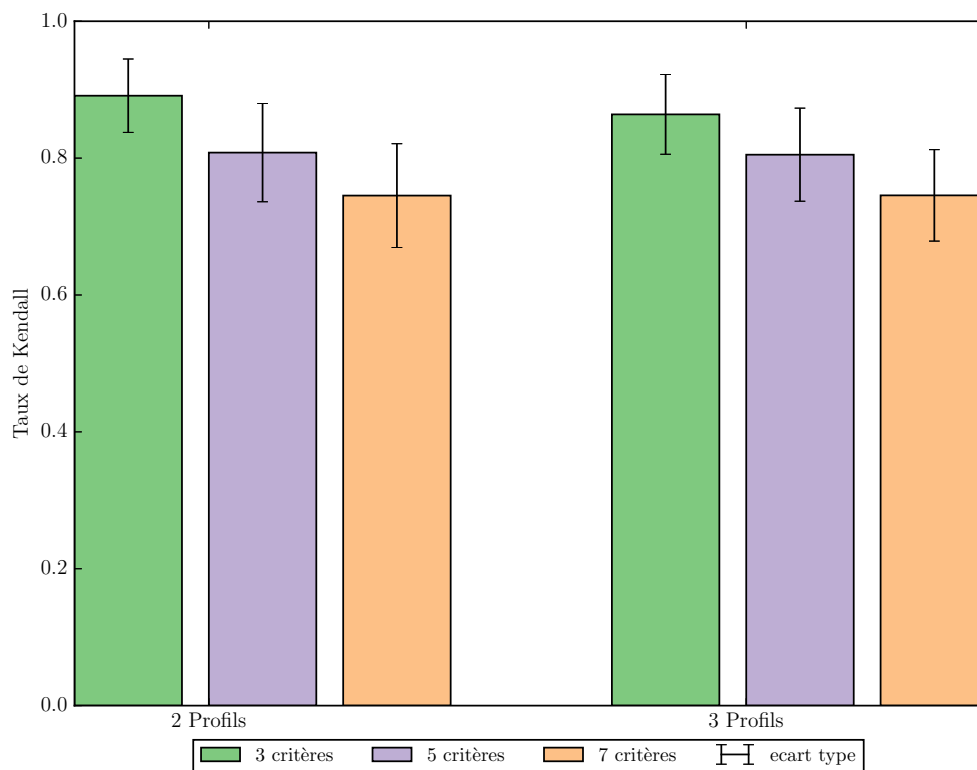


Figure 5.9 – Tau de *Kendall* moyen pour les différents problèmes

Nous remarquons que plus le problème est complexe c'est-à-dire plus le nombre de critères augmente plus la valeur du tau de *Kendall* diminue. En plus le nombre de profils a un effet sur le tau de *Kendall*, il est moins important pour les modèles SRMP à deux profils que ceux à trois profils. Nous observons ici sans surprise que plus le problème de décision est complexe, plus le nombre de paires d'apprentissage nécessaires augmente.

5.3 Résumé et mise en perspective

Pour éliciter les paramètres d'un modèle SRMP deux approches sont possibles. La première est l'approche directe où le décideur doit fixer les valeurs exactes de tous les paramètres, mais une telle approche est généralement difficile pour le décideur surtout s'il n'est pas habitué aux modèles de préférences de l'AMCD. La deuxième est l'approche indirecte qui consiste à éliciter les paramètres du modèle à partir de préférences holistiques.

Dans ce chapitre, nous focalisons sur l'élicitation indirecte des paramètres du modèle SRMP. Plus précisément sur l'élicitation par lot où la base des paires d'apprentissage complète est utilisée entièrement. En d'autres termes, les informations liées aux préférences du décideur sont

fournies en une seule itération.

Plusieurs modèles d'inférence existent dans la littérature pour éliciter les paramètres des modèles SRMP. Ils appartiennent à deux catégories : les méthodes exactes et les méthodes approchées. Nous commençons ce chapitre par présenter les modèles de la première catégorie, ces modèles permettent de trouver une solution optimale du problème d'élicitation des préférences.

Dans les méthodes exactes, on trouve notamment une approche par programmation linéaire mixte en nombres entiers (MIP). Ce programme est construit à partir d'une liste de comparaisons binaires indiquées par le décideur. Cette liste contient des paires de préférence stricte ou d'indifférence, qui génèrent des contraintes pour le MIP. Une approche basée sur un problème de satisfaisabilité booléenne (SAT) peut également être utilisée.

Nous proposons une étude d'expressivité des modèles SRMP. Le but de cette étude est de trouver le nombre de profils nécessaires pour garantir une bonne expressivité pour des problèmes avec 7 critères au maximum. Les résultats permettent de conclure que 3 profils au maximum sont nécessaires pour garantir une bonne expressivité et qu'ajouter des profils n'améliore cette expressivité qu'à la marge.

Les méthodes exactes pour inférer les paramètres d'un modèle SRMP nécessitent en général un temps de calcul élevé (avec des avantages significatifs pour le modèle basé sur la SAT). Cela limite leur utilisation dans des applications réelles, notamment quand la complexité des modèles à éliciter croît et que le nombre de comparaisons par paires devient élevé. Dans ces situations les méthodes approchées peuvent être utilisées pour réduire le temps de calcul et pouvoir traiter des plus grands problèmes avec un plus grand nombre de comparaisons binaires.

Nous proposons un algorithme génétique pour éliciter les paramètres d'un modèle SRMP. Différentes configurations et différents opérateurs sont testés pour trouver les mieux adaptés pour résoudre notre problème. Des tests supplémentaires sont faits pour évaluer le comportement de notre algorithme face à de nouvelles alternatives. Les résultats montrent un bon pouvoir de généralisation des modèles SRMP élicités par notre métaheuristique.

Dans le contexte de notre recherche doctorale, où nous proposons d'intégrer un modèle SRMP dans les décisions d'un drone autonome, il est important de configurer ce modèle selon les préférences de l'opérateur. Ce dernier doit donc être soumis à une phase d'élicitation des préférences en amont de la mission. Cependant, il est facilement imaginable qu'une élicitation par lot n'ait pas adapté à cette tâche, car elle nécessite un nombre élevé de comparaisons binaires, ce qui risque d'engendrer un grand effort cognitif pour l'opérateur (et qui risque par conséquent d'émettre des préférences non-compatibles avec le modèle élicité).

Pour répondre à ce problème de fatigue, nous attaquons dans le chapitre 6 le problème d'élicitation des modèles SRMP sous l'angle de l'élicitation incrémentale des préférences. Cette approche permet d'inférer le modèle SRMP de façon incrémentale et le décideur exprime ses préférences (préférence stricte ou indifférence) sur une paire d'apprentissage à chaque itération.

Chapitre 6

Élicitation incrémentale des paramètres de modèles SRMP

Sommaire

6.1 Élicitation incrémentale des préférences	110
6.1.1 Apprentissage actif et élicitation de préférences incrémentale	110
6.1.2 Processus d'élicitation incrémentale	111
6.1.3 Résolution des incohérences	114
6.2 Heuristiques de sélection	119
6.2.1 Heuristiques utilisant l'ensemble des alternatives	120
6.2.2 Heuristiques utilisant l'ensemble des solutions	120
6.3 Configuration du MIP	122
6.3.1 MIP_{first} : premier modèle SRMP réalisable	123
6.3.2 MIP_{close} : modèle SRMP le plus proche du modèle précédent	123
6.3.3 MIP_{center} : un modèle SRMP placé au centre de l'espace de recherche	124
6.4 Validation empirique	124
6.4.1 Plan d'expériences	125
6.4.2 Résultats expérimentaux et discussion	125
6.4.3 En pratique	133
6.5 Élicitation incrémentale avec l'algorithme génétique	134
6.6 Résumé et mise en perspective	138

Afin de réduire l'effort cognitif du décideur lors de la phase d'élicitation de ses préférences, nous proposons dans ce chapitre un processus d'élicitation incrémentale. Ce processus contient une étape de sélection des paires d'apprentissage. Nous étudions donc dans ce chapitre un ensemble d'heuristiques de sélection afin de trouver celle qui permet d'atteindre un modèle SRMP satisfaisant avec le minimum de paires d'apprentissage. En effet, le but de ces heuristiques est de sélectionner à chaque itération du processus incrémental une paire d'alternatives et de la proposer au décideur pour exprimer ses préférences. Ces dernières sont transformées en des contraintes d'un modèle d'inférence pour la construction d'un modèle SRMP. Pour notre étude nous démarrons avec un programme linéaire mixte en nombres entiers (MIP) comme modèle d'inférence qui permet de trouver une solution exacte. Différentes configurations de MIP sont présentées dans ce chapitre dont le but est de choisir le bon modèle SRMP parmi l'ensemble des modèles compatibles. L'étude des différentes heuristiques se fait avec les différentes configurations de MIP afin

de trouver la meilleure combinaison d’heuristique et de configuration de MIP. Les résultats de cette étude permettent de choisir une heuristique de choix de paires et une configuration du MIP. Cependant, notre étude montre également que les temps de calcul sont trop élevés pour une élicitation interactive avec un décideur. Cela nous mène à tester l’intégration de la métaheuristique proposée dans le chapitre 5 dans le processus incrémental, en vue de rendre le processus plus réactif. Nous montrons qu’il faut accepter un compromis entre le temps de calcul et la qualité de la solution espérée.

Les travaux présentés dans ce chapitre s’inspirent de notre article [Khannoussi *et al.*, 2018b] et sont repris dans un article en cours de révision pour une publication dans la revue 4OR. Ils ont aussi fait l’objet d’une présentation à la 88ème réunion du Groupe de travail EURO sur l’Aide Multi-Critère à la Décision [Khannoussi *et al.*, 2018c]

6.1 Élicitation incrémentale des préférences

Nous nous intéressons ici à l’*élicitation incrémentale* des préférences pour les modèles SRMP. Cette approche incrémentale est inspirée par le domaine d’apprentissage automatique (machine learning), plus précisément de l’*apprentissage actif*. Nous détaillons dans la suite ces deux termes et leurs utilisations dans le domaine d’apprentissage automatique et d’AMCD.

6.1.1 Apprentissage actif et élicitation de préférences incrémentale

Classiquement, les méthodes d’apprentissage automatique (machine learning) sont définies comme des algorithmes qui déduisent des informations structurelles à partir des données. Certaines de ces techniques se limitent à l’apprentissage par lot, où toutes les données sont fournies avant l’apprentissage. Cependant, dans de nombreuses applications, les données arrivent par un flux constant, ce qui nécessite une adaptation continue du modèle. L’approche utilisée pour ce contexte est appelée *apprentissage actif*.

L’apprentissage actif se focalise sur l’arrivée séquentielle des données et sur l’utilisation d’un oracle pendant le processus. Le terme actif est inspiré par le fait que l’algorithme d’apprentissage peut influencer sur le choix des exemples d’apprentissage. Plus précisément, contrairement au cadre traditionnel où les données sont connues et imposées, dans l’apprentissage actif, c’est l’algorithme d’apprentissage qui demande des informations pour des données spécifiques. Ces algorithmes reçoivent les données d’apprentissage séquentiellement, un par un ou lot par lot, et l’utilisent, ainsi que le modèle précédemment appris et l’entrée de l’oracle pour en produire un meilleur modèle. Ils se présentent sous trois formes dans la littérature : génération d’instances, apprentissage en ligne et hors ligne. Dans le premier type d’algorithmes, les nouvelles instances à étiqueter sont générées par l’algorithme, alors que dans le second type, les données d’apprentissage arrivent séquentiellement, et l’algorithme d’apprentissage décide s’il est nécessaire ou non d’être confronté à l’oracle. Les algorithmes d’apprentissage hors ligne supposent qu’il existe une base de données d’instances, et l’algorithme sélectionne quelles données sont confrontées aux connaissances de l’oracle. Quelques exemples de l’utilisation de techniques d’apprentissage actif peuvent être trouvés pour la construction incrémentale d’arbres de décision pour la classification ou la régression [Fan *et al.*, 2015, Gholipour *et al.*, 2013, Salperwyck et Lemaire, 2013] et plusieurs modèles SVM incrémentaux sont présentés dans [Wen et Lu, 2007]. [Syed *et al.*, 2017] présentent une approche pour l’apprentissage actif dans les réseaux neuronaux convolutionnels profonds.

Dans notre contexte, où l’algorithme de décision est un modèle d’AMCD, le sujet de l’apprentissage actif a surtout été abordé du point de vue hors ligne, et s’appelle habituellement « *élicitation incrémentale des préférences* ». Dans le contexte de MAVT, [Durbach, 2009] et [Lahdelma *et al.*, 1998] utilisent un indice qui quantifie le volume du polyèdre des contraintes précisant les fonctions de valeur possibles et tentent de réduire ce volume en ajoutant des contraintes représentant la comparaison par paire des alternatives, jusqu’à ce que celles-ci convergent vers la meilleure solution. [Holloway et White III, 2003] montrent l’importance de l’ordre des comparaisons par paires pour diminuer le nombre de questions afin de réduire l’effort cognitif du décideur. [Ciomek *et al.*, 2017] présentent un ensemble d’heuristiques pour minimiser le nombre de questions d’élicitation et les classer par ordre de priorité dans le contexte des problèmes de décision à choix unique. Ils concluent que l’heuristique la plus performante dépend des paramètres du problème (par exemple nombre de critères et d’alternatives). Dans le même contexte, [Benabbou *et al.*, 2017] sélectionnent une série de comparaisons par paire en utilisant une stratégie de regret minimal. Cette stratégie réduit ce nombre, mais la performance est affaiblie. En outre, Olteanu [Olteanu, 2018] présente des stratégies pour générer des alternatives pour l’obtention incrémentale de modèles de tri du type MR-Sort.

Inspirés par le domaine de l’apprentissage machine et les travaux mentionnés précédemment en AMCD, nous choisissons d’étudier dans ce travail l’élicitation incrémentale des modèles SRMP afin de réduire l’effort cognitif du décideur pendant le processus d’élicitation des préférences. Nous proposons en premier temps un processus d’élicitation incrémentale adapté à notre objectif. Ensuite, nous proposons des heuristiques pour sélectionner des paires d’alternatives à partir d’une base de données existante, que le décideur doit comparer. À notre connaissance, aucun travail antérieur ne traite de l’élicitation incrémentale des modèles de classement issus du paradigme du surclassement, et plus particulièrement du modèle de classement SRMP.

6.1.2 Processus d’élicitation incrémentale

Le processus d’élicitation incrémentale que nous proposons est illustré dans la figure 6.1. Comme mentionné précédemment, nous avons à notre disposition une base de paires d’alternatives existantes qui est présentée dans la figure 6.1 en haut à gauche. Cette base de données constitue la source de données de notre processus d’élicitation. Nous sommes donc bien dans un processus hors ligne, comme décrit ci-avant, où une base de données d’apprentissage existe en amont du processus, et où un algorithme va sélectionner à chaque étape la paire d’alternatives à proposer à un oracle (le décideur dans ce cas). À chaque itération, une heuristique sélectionne une paire d’alternatives dans la base de données qui sera proposée au décideur. Il exprime ensuite ses préférences sur cette paire d’alternatives (indifférence ou préférence stricte), ce qui génère une nouvelle comparaison binaire qui s’ajoute aux précédentes. Ces comparaisons sont utilisées comme entrées dans le modèle d’inférence pour construire un nouveau modèle SRMP. Cette procédure est répétée, comme le montre la flèche qui revient à la sélection de la prochaine paire. Ce processus interactif s’arrête lorsque le décideur est satisfait du modèle élicité, ou lorsqu’un critère d’arrêt est vérifié.

Le choix d’avoir une base de paires d’alternatives existante s’est imposé par notre contexte des drones. En effet, le drone doit choisir une action de haut niveau (atterrir, continuer, ...) à exécuter lorsque le niveau 2 du moteur que nous proposons est déclenché. À chaque fois que cette décision doit être exécutée, seulement un nombre limité d’alternatives sont réalisables. Les paires d’alternatives présentes dans la base tiennent donc compte de cette contrainte d’existence des alternatives, et sont construites de façon à ce que les 2 alternatives de chaque paire sont

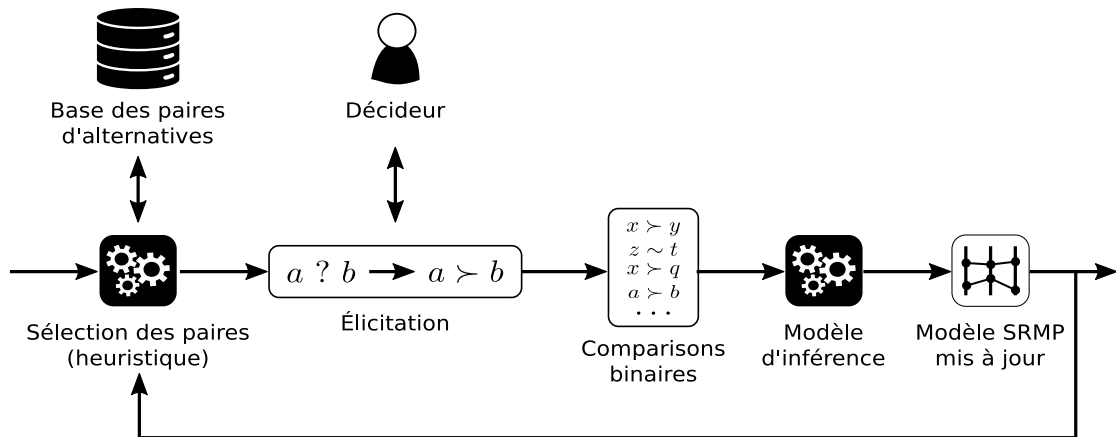


Figure 6.1 – Processus d'élicitation incrémentale général

bien compatibles dans un contexte donné. Donc les paires d'alternatives présentes dans la base sont faites de sorte que les deux actions sont réalisables dans un même waypoint. Une autre raison importante qui nous a guidés à ce choix est le fait qu'une action de haut niveau peut être exécutée dans différents contextes, mais son évaluation selon les critères retenus varie d'un contexte à l'autre. Donc au cours du processus incrémental le décideur peut rencontrer plusieurs fois la même action, mais avec des évaluations différentes d'une paire à une autre.

Par exemple, si nous voulons demander au pilote d'exprimer ses préférences sur deux actions, nous devons en premier temps lui présenter le contexte et l'état du drone (chapitre 3). Pour cet exemple nous considérons que le drone survole la mer avec la détection d'un changement météorologique qui exige une modification du plan de mission en prenant une décision de haut niveau. Dans ce cas nous ne pouvons pas demander à l'opérateur d'exprimer ses préférences sur la paire d'actions « atterrir » et « continuer » parce que l'action « atterrir » n'est pas réalisable dans ce contexte vu que le drone survole la mer. Mais nous pouvons lui présenter la paire « continuer » et « sauter un waypoint ». À l'itération suivante les deux mêmes actions « continuer » et « sauter un waypoint » peuvent être présentées à l'opérateur, mais avec des évaluations différentes puisque le contexte n'est plus le même qu'avant. Cela explique la présence d'une base de paires d'alternatives dans notre processus.

Le processus itératif s'arrête lorsque nous obtenons un modèle de préférence « satisfaisant ». Il est évident que le terme « satisfaisant » peut avoir des définitions différentes, selon l'application ou le décideur. Dans ce chapitre, nous donnons quelques conseils sur la façon de déterminer dans la pratique quand arrêter le processus. La qualité de ce modèle de préférences dépend du nombre d'itérations du processus d'élicitation précédent, de la complexité du modèle SRMP (déterminée par le nombre de profils utilisés) et de la structure du problème décisionnel (généralement le nombre de critères). Dans la section 6.4, l'un de nos objectifs est de déterminer le nombre d'itérations pour ce processus afin de garantir, en moyenne, un modèle de préférence d'une qualité donnée.

Ce processus a l'avantage d'être générique ce qui veut dire que le modèle d'inférence peut être un parmi les algorithmes d'élicitation du modèle SRMP présentés dans le chapitre 3. Nous pouvons utiliser ce processus avec des algorithmes à solutions exactes comme le MIP [Olteanu

et al., 2018] ou le SAT [Belahcene *et al.*, 2018] ou à solutions approchées comme la métaheuristique présentée par [Liu *et al.*, 2014] ou celle que nous avons proposée dans la section 5.2.

Dans des cas d'application réels, un contexte d'informations imprécises [Weber, 1987, Dias et Clímaco, 2000] peut apparaître. Ce contexte est lié soit à l'incertitude des préférences du décideur, à l'incompatibilité du modèle SRMP avec les préférences du décideur ou à des données imprécises. Pour le premier scénario, le décideur exprime ses préférences sur une nouvelle paire d'alternatives, mais elles se révèlent incompatibles avec les préférences exprimées précédemment. Dans le deuxième scénario, le décideur n'exprime pas ses préférences selon le modèle SRMP. Pour illustrer cela, considérons un décideur qui de manière intuitive applique un modèle compensatoire lorsqu'il doit exprimer des préférences entre deux alternatives. Ceci n'est pas nécessairement compatible avec un modèle SRMP, et cela montre donc l'importance de suivre le processus de l'aide à décision présenté dans le chapitre 3 pour bien choisir le modèle d'AMCD à utiliser. Dans la suite de notre travail, nous nous restreignons au cas d'incompatibilité.

Focalisons-nous sur le cas où à une itération donnée les préférences exprimées par le décideur sur une paire d'alternatives ne sont pas compatibles avec les préférences exprimées aux itérations précédentes. Il est alors important de résoudre ce problème afin de poursuivre le processus d'élicitation incrémentale et avoir à la fin un modèle SRMP compatible avec les préférences du décideur.

Dans le cas où un programme linéaire mixte en nombres entiers (MIP) est utilisé pour inférer le modèle SRMP, résoudre l'incompatibilité des contraintes revient à demander au décideur de supprimer une ou plusieurs comparaisons binaires. Cela complète notre processus d'élicitation incrémentale comme le montre la figure 6.2. S'il n'y a pas de solution suite à l'exécution du modèle d'inférence (MIP) le décideur intervient dans la partie de résolution des incohérences pour choisir parmi les sous-ensembles de paires possibles lesquelles supprimer. Cela produit une nouvelle liste de comparaisons binaires qui permet d'inférer un modèle SRMP en respectant la nouvelle liste des contraintes (sans incohérence).

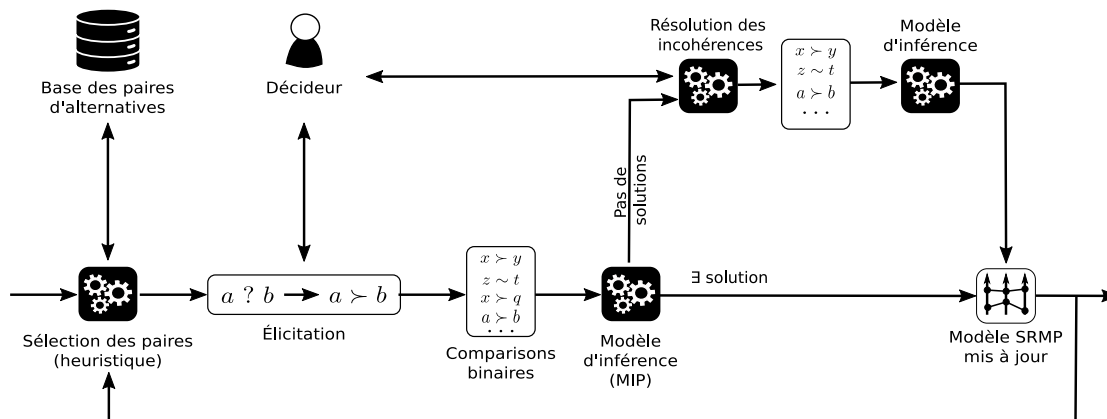


Figure 6.2 – Processus d'élicitation incrémentale en cas d'incohérence de préférences

Dans la suite nous présentons le processus détaillé à suivre pour trouver la liste des sous-ensembles des paires possibles à présenter au décideur afin qu'il puisse choisir lequel supprimer.

6.1.3 Résolution des incohérences

[Mousseau *et al.*, 2003] présentent une approche pour résoudre le problème d'incohérence des contraintes pour les modèles d'AMCD où un programme linéaire est utilisé comme modèle d'inférence. Ils considèrent un problème dans lequel le décideur spécifie de façon interactive les contraintes sur les paramètres d'un modèle d'AMCD qui définissent un polyèdre de solutions possibles. À l'itération $k - 1$ ce polyèdre (noté T_{k-1}) est généralement défini par un système (noté $S(1)$) composé de $m - 1$ contraintes linéaires sur n variables x_1, x_2, \dots, x_n :

$$S(1) = \begin{cases} \sum_{j=1}^n \alpha_{1,j} x_j & \geq \beta_1, \\ & \vdots \\ \sum_{j=1}^n \alpha_{m-1,j} x_j & \geq \beta_{m-1}, \end{cases} \quad \alpha_{i,j}, \beta_i \in \mathbb{R}, i = 1, \dots, m-1, j = 1, \dots, n.$$

À l'itération k , le décideur ajoute une nouvelle contrainte, $\sum_{j=1}^n \alpha_m x_j \geq \beta_m$ qui est ajoutée à $S(1)$ pour créer un nouveau système $S(2)$ qui contient m contraintes linéaires :

$$S(2) = \begin{cases} \sum_{j=1}^n \alpha_{1,j} x_j & \geq \beta_1, \\ & \vdots \\ \sum_{j=1}^n \alpha_{m-1,j} x_j & \geq \beta_{m-1}, \\ \sum_{j=1}^n \alpha_{m,j} x_j & \geq \beta_m \end{cases} \quad \alpha_{i,j}, \beta_i \in \mathbb{R}, i = 1, \dots, m-1, j = 1, \dots, n.$$

Cette nouvelle contrainte est incompatible avec les précédentes ce qui rend le système $S(2)$ incohérent. Le décideur doit choisir entre supprimer la nouvelle contrainte ou supprimer certaines des anciennes. Les auteurs proposent d'identifier des sous-ensembles de contraintes qui, une fois supprimés, conduisent à un système cohérent. Identifier de tels sous-ensembles indiquera la raison de l'incohérence de l'information fournie par le décideur. Il peut y avoir plusieurs possibilités pour le décideur pour résoudre l'incohérence.

Ils notent $\mathcal{I} = \{1, 2, \dots, m\}$ les indices des contraintes à l'itération k avec la nouvelle contrainte incompatible où $T_k = \{x \in \mathbb{R}^n : \sum_{j=1}^n \alpha_{i,j} x_j \geq \beta_i, \forall i \in \mathcal{I}\} = \emptyset$. $S_t \subset \mathcal{I}$ est un sous-ensemble d'indices des contraintes. S_t résout le $S(2)$ si seulement si $\sum_{j=1}^n \alpha_{i,j} x_j \geq \beta_i \forall i \in \mathcal{I} \setminus S_t$. Les auteurs cherchent à identifier tous les sous-ensembles minimaux S_t qui résolvent $S(2)$ avec une cardinalité inférieure ou égale à Ω (définie à l'avance). Ils déterminent le p ensembles de $S_t, t = 1, \dots, p$ s'ils existent, de sorte que :

- (i) S_t résout $S(2), t \in \{1, \dots, p\}$;
- (ii) $S_t \not\subseteq S_s, t, s \in \{1, \dots, p\}, t \neq s$;
- (iii) $|S_t| \leq |S_s|, t, s \in \{1, \dots, p\}, t < s$;
- (iv) S'il existe un sous-ensemble S_s d'indices de contraintes qui résout $S(2)$ tel que $S_s \not\subseteq S_t, \forall t \in \{1, \dots, p\}$, alors $|S_s| > |S_t|$;
- (v) $|S_i| \leq \Omega \forall i \in \{1, 2, \dots, p\}$.

Les auteurs proposent deux algorithmes différents pour trouver les différents sous-ensembles. Le premier est basé sur les techniques de la programmation linéaire mixtes à variables binaire et

le deuxième qui est basé sur les techniques de la programmation linéaire.

Nous détaillons ici le premier algorithme qui sera appliqué à l'inférence du modèle SRMP dans la suite. L'identification des p sous-ensembles de contraintes est faite à travers $p - 1$ optimisations successives $(PM_2, PM_3, \dots, PM_p)$. Le programme PM_2 permet de minimiser le nombre de contraintes à supprimer afin de rendre T_k faisable. Le sous-ensemble $S_1 = \{m\}$ est évidemment le plus petit sous-ensemble vérifiant (i)-(v). Le premier problème à résoudre PM_2 est le suivant :

$$PM_2 = \begin{cases} \text{Min} & \sum_{i=1}^m Y_i \\ \text{s.t.} & \sum_{j=1}^n \alpha_{i,j} x_j + MY_i \geq \beta_i, \forall i \in \mathcal{I} \setminus \{m\}, \\ & \sum_{j=1}^n \alpha_m x_j + MY_i \geq \beta_m, \\ & x_j \leq 0, j \in 1, \dots, n, \\ & Y_i \in \{0, 1\}, \forall i \in \mathcal{I} \setminus \{m\}. \end{cases}$$

M est un grand nombre positif et les variables $Y_i, \forall i \in \mathcal{I} \setminus \{m\}$ sont des variables binaires associées à chaque contrainte, elles permettent d'activer ou de désactiver les contraintes selon que leur valeur est égale à 1 ou 0. À l'optimum de PM_2 , les indices des contraintes où $Y_i^* = 1$ constituent le sous-ensemble S_2 .

Pour trouver le sous-ensemble S_3 il faut résoudre le problème PM_3 qui n'est que le PM_2 avec une seule contrainte de plus $\sum_{i \in S_2} Y_i \leq |S_2| - 1$. Cette contrainte empêche la PM_3 de trouver une solution optimale qui corresponde à (ou inclus) S_2 . Les indices des contraintes où $Y_i^* = 1$ (PM_3 est optimal) constituent le sous-ensemble S_3 . Pour trouver le reste des sous-ensembles S_4, \dots, S_p , il faut procéder de la même façon. Chaque nouveau programme est formé par l'ajout d'une contrainte au programme précédent. L'algorithme 5 détaille les étapes à suivre pour trouver $p - 1$ sous-ensembles de contraintes.

Algorithme 5 Calcul des sous-ensembles des contraintes

```

k := 2
plusSol := vrai
tant que plusSol faire
  si ( $PM_k$  n'a pas de solution) ou ( $PM_k$  a une valeur optimale  $\geq \Omega$ ) alors
    plusSol := faux
  sinon
     $S_k := \{i \in \mathcal{I} \setminus \{m\}\}$ 
    Ajouter la contrainte  $\sum_{i \in S_k} Y_k \leq |S_k| - 1$ 
    k := k + 1
  fin si
fin tant que

```

[Mousseau *et al.*, 2003] considèrent que les préférences données par le décideur à chaque itération génère une seule contrainte linéaire. Dans notre cas, le décideur exprime ses préférences (préférence stricte ou indifférence) sur une paire d'alternatives qui se rajoute à liste des comparaisons binaires. En utilisant le MIP présenté dans la section 5.1.1 une nouvelle comparaison binaire génère un ensemble de contraintes linéaires. Notre but est de trouver les sous-ensembles des paires d'alternatives possibles à supprimer pour résoudre l'incohérence plutôt que les sous-

<i>Paramètres :</i>	
$\mathcal{A}, \mathcal{Pr}, \mathcal{In}, M, k, \sigma, \gamma, \theta$	(1)
$a_j \in [0, 1]$	$\forall a \in \mathcal{A}, \forall j \in M$ (2)
<i>Variables :</i>	
$w_j \in [0, 1]$	$\forall j \in M$ (3)
p_j^h	$\forall h \in 1..k, \forall j \in M$ (4)
$\delta_j(a, p^h) \in \{0, 1\}$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M$ (5)
$\omega_j(a, p^h) \in [0, 1]$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M$ (6)
$s(a, b, p^h) \in \{0, 1\}$	$\forall (a, b) \in \mathcal{Pr}, \forall h \in 0..k$ (7)
$Y_{\mathcal{Pr}}(a, b) \in \{0, 1\}$	$\forall (a, b) \in \mathcal{Pr}$ (8)
$Y_{\mathcal{In}}(a, b) \in \{0, 1\}$	$\forall (a, b) \in \mathcal{In}$ (9)
<i>Objectif :</i>	
$\text{minimiser} : \sum Y_{\mathcal{Pr}}(a, b) + \sum Y_{\mathcal{In}}(a', b')$	$\forall (a, b) \in \mathcal{Pr}, \forall (a', b') \in \mathcal{In}$ (10)
<i>Contraintes :</i>	
$s.t. \sum_{j \in M} w_j = 1$	(11)
$w_j \geq \gamma$	$\forall j \in M$ (12)
$p_j^1 \geq 0$	$\forall j \in M$ (13)
$1 \geq p_j^k$	$\forall j \in M$ (14)
$p_j^{h+1} \geq p_j^h$	$\forall h \in 1..k-1, \forall j \in M$ (15)
$a_j - p_j^h \geq \delta_j(a, p^h) - 1$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M$ (16)
$\delta_j(a, p^h) \geq a_j - p_j^h + \gamma$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M$ (17)
$w_j \geq \omega_j(a, p^h) - \theta * Y_{\mathcal{Pr}}(a, b)$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M, \forall (a, b) \in \mathcal{Pr}$ (18)
$w_j \geq \omega_j(a, p^h) - \theta * Y_{\mathcal{In}}(a, b)$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M, \forall (a, b) \in \mathcal{In}$ (19)
$\omega_j(a, p^h) \geq 0 - \theta * Y_{\mathcal{Pr}}(a, b)$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M, \forall (a, b) \in \mathcal{Pr}$ (20)
$\omega_j(a, p^h) \geq 0 - \theta * Y_{\mathcal{In}}(a, b)$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M, \forall (a, b) \in \mathcal{In}$ (21)
$\delta_j(a, p^h) \geq \omega_j(a, p^h) - \theta * Y_{\mathcal{Pr}}(a, b)$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M, \forall (a, b) \in \mathcal{Pr}$ (22)
$\delta_j(a, p^h) \geq \omega_j(a, p^h) - \theta * Y_{\mathcal{In}}(a, b)$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M, \forall (a, b) \in \mathcal{In}$ (23)
$\omega_j(a, p^h) \geq \delta_j(a, p^h) + w_j - 1 - \theta * Y_{\mathcal{Pr}}(a, b)$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M, \forall (a, b) \in \mathcal{Pr}$ (24)
$\omega_j(a, p^h) \geq \delta_j(a, p^h) + w_j - 1 - \theta * Y_{\mathcal{In}}(a, b)$	$\forall a \in \mathcal{A}, \forall h \in 1..k, \forall j \in M, \forall (a, b) \in \mathcal{In}$ (25)
$s(a, b, p^0) = 1$	$\forall (a, b) \in \mathcal{Pr}$ (26)
$s(a, b, p^k) = 0$	$\forall (a, b) \in \mathcal{Pr}$ (27)
$\sum_{j \in M} \omega_j(a, p^{\sigma(h)}) \geq \sum_{j \in M} \omega_j(b, p^{\sigma(h)}) + \gamma - s(a, b, p^{\sigma(h)}) \cdot (1 + \gamma) - (1 - s(a, b, p^{\sigma(h-1)})) - \theta * Y_{\mathcal{Pr}}(a, b)$	$\forall (a, b) \in \mathcal{Pr}, \forall h \in 1..k$ (28)
$\sum_{j \in M} \omega_j(a, p^{\sigma(h)}) \geq \sum_{j \in M} \omega_j(b, p^{\sigma(h)}) - (1 - s(a, b, p^{\sigma(h)})) - (1 - s(a, b, p^{\sigma(h-1)})) - \theta * Y_{\mathcal{Pr}}(a, b)$	$\forall (a, b) \in \mathcal{Pr}, \forall h \in 1..k$ (29)
$\sum_{j \in M} \omega_j(b, p^{\sigma(h)}) + (1 - s(a, b, p^{\sigma(h)})) + (1 - s(a, b, p^{\sigma(h-1)})) \geq \sum_{j \in M} \omega_j(a, p^{\sigma(h)}) - \theta * Y_{\mathcal{Pr}}(a, b)$	$\forall (a, b) \in \mathcal{Pr}, \forall h \in 1..k$ (30)
$\sum_{j \in M} \omega_j(a, p^{\sigma(h)}) \geq \sum_{j \in M} \omega_j(b, p^{\sigma(h)}) - \theta * Y_{\mathcal{In}}(a, b)$	$\forall (a, b) \in \mathcal{In}, \forall h \in 1..k$ (31)
$\sum_{j \in M} \omega_j(a, p^{\sigma(h)}) - \theta * Y_{\mathcal{In}}(a, b) \leq \sum_{j \in M} \omega_j(b, p^{\sigma(h)})$	$\forall (a, b) \in \mathcal{In}, \forall h \in 1..k$ (32)
$Y_{\mathcal{Pr}}(x, y) = 1$	(33)

Tableau 6.1 – MP_2 pour déterminer les contraintes incompatibles dans l'inférence d'un modèle SRMP

ensembles des contraintes.

Afin de trouver les sous-ensembles de paires d'alternatives à présenter au décideur pour qu'il puisse choisir lesquelles supprimer, nous associons des variables binaires à chaque ensemble de contraintes généré par chaque comparaison binaire. Puisque nous avons des paires de préférences et d'indifférences nous introduisons deux ensembles des variables binaires $Y_{Pr}(a, b), \forall(a, b) \in Pr$ et $Y_{In}(a, b), \forall(a, b) \in In$.

Inspirés par le travail de [Mousseau *et al.*, 2003], nous définissons le programme PM_2^{SRMP} adapté à notre cas. Le tableau 6.1 montre que le PM_2^{SRMP} permet de minimiser le nombre des paires d'alternatives à supprimer afin d'avoir un modèle SRMP possible (10). Les contraintes (18), (20), (22),(24), (28),(29),(30) sont générées par les paires de préférences alors que les (19), (21), (23),(24), (31),(32) sont liées à la liste des paires d'indifférences.

Nous supposons ici que la dernière paire d'alternatives (x, y) qui a généré l'incohérence est une préférence c'est çà dire $(x, y) \in Pr$. Un argumentaire similaire peut être déroulé avec $(x, y) \in In$. Le décideur pourrait changer d'avis sur cette dernière paire ajoutée dans le processus d'élicitation incrémentale. Cependant, nous considérons qu'il souhaite la maintenir. Il s'agit donc maintenant de déterminer des sous-ensembles de paires à enlever, afin de rendre le système cohérent. Il en découle que nous fixons $Y_{Pr}(x, y) = 1$ (33).

La résolution de ce programme donne $S_2 = S_2^{Pr} \cup S_2^{In}$ comme solution optimale avec $S_2^{Pr} = \{\forall(a, b) \in Pr \setminus Y_{Pr}(a, b) = 1\}$ et $S_2^{In} = \{\forall(a, b) \in In \setminus Y_{In}(a, b) = 1\}$. En suivant l'algorithme 5, le PM_3^{SRMP} est défini en rajoutant à PM_2^{SRMP} la contrainte $\sum_{(a,b) \in S_2^{Pr}} Y_{Pr}(a, b) + \sum_{(a,b) \in S_2^{In}} Y_{In}(a, b) \leq |S_2^{Pr}| + |S_2^{In}| - 1$. De la même façon, nous déterminons le reste des sous-ensembles jusqu'à arriver à un programme PM_k^{SRMP} sans solution réalisable.

Exemple illustratif :

Nous considérons dans cet exemple que nous avons un problème de 3 critères et le nombre de profils du modèle SRMP est fixé à 2. La liste des paires des préférences et d'indifférences est présentée dans le tableau 6.2. Le tableau 6.3 présente les évaluations de l'ensemble des 20 alternatives. Jusqu'à l'itération 14 il existe bien un modèle SRMP qui représente les préférences du décideur.

nb itération	Préférences	Indifférences
1	b \succ j	
2	j \succ f	
3	t \succ s	
4	s \succ d	
5	a \succ l	
6	i \succ o	
7	o \succ e	
8	e \succ h	
9	q \succ c	
10	p \succ r	
11	n \succ k	
12	k \succ m	
13		g \sim q
14		r \sim n
15	d \succ a	

Tableau 6.2 – Liste des préférences/indifférences du décideur

alternatives	cr1	cr2	cr3
a	85.24205825	93.73806137	89.35218133
b	18.66818824	36.731212	96.25284414
c	8.65796418	95.63122168	60.80411962
d	86.07453331	24.5670322	63.58786638
e	84.5392806	13.61462665	93.62910559
f	99.56995833	73.06492279	29.01188849
g	63.27260102	67.92315001	91.69831582
h	53.37632201	94.72992475	82.92774544
i	7.98454009	84.92364292	92.01274348
j	74.2369528	13.38539699	99.16344085
k	87.83814027	17.93565327	53.33654532
l	6.42327897	97.18118934	4.97009635
m	91.17771469	16.97654121	64.92985475
n	91.40646102	13.68952693	76.21443891
o	85.41224571	35.87119128	86.8906473
p	32.43436177	89.11504263	91.51763793
q	22.17515827	26.55367291	94.19018698
r	29.75157239	94.77129272	64.06343335
s	97.48937199	55.11980603	48.49309943
t	44.80543502	49.66995311	94.00136341

Tableau 6.3 – Évaluations des alternatives

À l'itération 15, nous présentons la paire (d, a) au décideur et il trouve que l'alternative d est aussi moins bonne que l'alternative a , donc cette paire est ajoutée à la liste des préférences \mathcal{P}_r . Mais l'ajout de cette paire a causé une incohérence dans le modèle d'inférence. Donc nous allons essayer de trouver les sous-ensembles des paires d'alternatives possibles à supprimer pour résoudre cette incohérence.

Nous calculons, étape par étape, toutes les solutions réalisables pour l'exemple sachant que nous nous restreignons à des sous-ensembles composés d'au plus 3 paires d'alternatives (c.-à-d. $\Omega = 4$) :

1. $S_1 = \{(d, a)\}$ (évident)
2. Nous obtenons $S_2^{Pr} = \{(s, d), (k, m)\}$ et $S_2^{In} = \emptyset$ donc $S_2 = \{(s, d), (k, m)\}$ comme solution pour PM_2^{SRMP} . La contrainte $Y_{Pr}(s, d) + Y_{Pr}(k, m) \leq 1$ est ajoutée au PM_2^{SRMP} .
3. La solution $S_3 = \{(s, d), (n, k)\}$ de PM_3^{SRMP} . La contrainte $Y_{Pr}(s, d) + Y_{Pr}(n, k) \leq 1$ est ajoutée au PM_3^{SRMP} .
4. La solution $S_4 = \{(a, l), (n, k), (o, e)\}$ de PM_4^{SRMP} . La contrainte $Y_{Pr}(a, l) + Y_{Pr}(n, k) + Y_{Pr}(o, e) \leq 2$ est ajoutée au PM_4^{SRMP} .
5. La solution $S_5 = \{(t, s), (n, k), (e, h)\}$ de PM_4^{SRMP} . La contrainte $Y_{Pr}(t, s) + Y_{Pr}(n, k) + Y_{Pr}(e, h) \leq 2$ est ajoutée au PM_4^{SRMP} .

6. Le problème devient infaisable ce qui signifie qu'il n'y a plus de sous-ensembles à proposer.

La liste des sous-ensembles des paires d'alternatives à proposer au décideur est donc la suivante :

$$\{(d, a)\}, \{(s, d), (k, m)\}, \{(s, d), (n, k)\}, \{(t, s), (n, k), (e, h)\}$$

Le décideur trouve que la paire (d, a) est plus importante par rapport à (s, d) . Cette dernière est incluse dans deux sous-ensembles $\{(s, d), (k, m)\}$ et $\{(s, d), (n, k)\}$ mais il préfère garder (k, m) également. Donc il décide de supprimer le sous-ensemble $\{(s, d), (n, k)\}$. Cela permet de résoudre le problème d'incohérence et continuer le processus d'élicitation incrémentale.

Dans la suite, nous focalisons sur la partie sélection des paires d'alternatives de la base existantes. Nous proposons et étudions un ensemble d'heuristiques de sélection pour en choisir une qui permet de réduire le plus la fatigue cognitive du décideur, autrement dit réduire le nombre des paires d'alternatives. Afin de faciliter l'étude de ces heuristiques nous considérons qu'il n'y a pas d'incohérence dans les informations données par le décideur. Dans la section qui suit, nous reprenons le processus d'élicitation générale comme présenté dans la figure 6.1.

6.2 Heuristiques de sélection des paires d'apprentissage

Pour la sélection de la paire à intégrer dans le processus d'élicitation incrémentale, nous proposons d'étudier différentes *heuristiques de sélection*, que nous divisons en deux catégories. Le premier groupe d'heuristiques se concentre exclusivement sur l'espace de recherche des alternatives, c'est-à-dire sans utiliser le modèle généré dans l'itération précédente lors de la sélection de la paire suivante. Le second groupe est lié à l'espace de recherche des solutions du modèle SRMP (et donc aux paramètres de préférence) en considérant le modèle déduit lors de l'itération précédente lors du choix de la paire suivante d'alternatives. Ceci est représenté par la flèche discontinue dans la figure 6.3 qui montre que le modèle SRMP précédent est utilisé dans la sélection de la paire d'alternatives suivante.

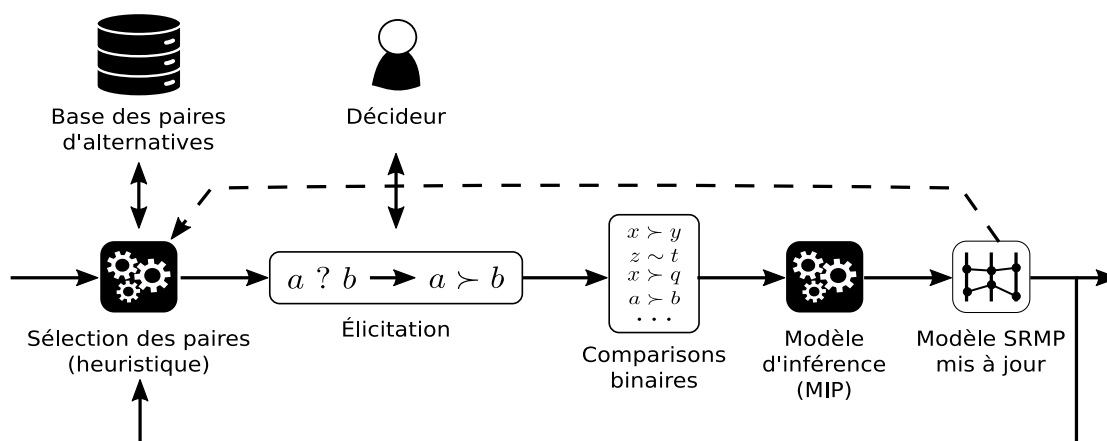


Figure 6.3 – Processus d'élicitation incrémentale

Nous supposons ici que la complexité du modèle recherché est fixée à l'avance, c'est-à-dire le

nombre de profils à utiliser pour le modèle SRMP, ainsi que leur ordre lexicographique. Dans les problèmes de décision réels, le modèle SRMP permet d'évaluer les alternatives sur les critères de manière qualitative ou quantitative. Cependant, pour faciliter notre discours dans ce travail, ainsi que les expériences de la section 6.4, nous supposons ici, sans perte de généralité, que toutes les évaluations sont sur l'échelle numérique $\mathcal{X}_j = [0, 1], \forall j \in \mathcal{M}$, et que tous les critères doivent être maximisés. Les heuristiques proposées peuvent facilement être adaptées au cas où les critères ont des échelles hétérogènes.

6.2.1 Heuristiques utilisant l'ensemble des alternatives

6.2.1.1 \mathcal{H}_{rnd} : heuristique aléatoire

Cette heuristique appartient au premier groupe et sélectionne simplement de façon séquentielle et aléatoire une paire d'alternatives dans l'ensemble $\mathcal{D} = \{(a, b) | a, b \in \mathcal{A}, a \neq b\}$, c'est-à-dire la base de données contenant les paires possibles.

$$(a, b)_{\text{rnd}} = \text{random}(\mathcal{D})$$

Avec cette heuristique nous espérons qu'un choix aléatoire permettra d'explorer l'espace des paires possibles le plus largement possible, et que cette heuristique nous servira aussi de référence.

6.2.1.2 \mathcal{H}_{sim} : heuristique d'alternatives similaires

À l'aide de cette heuristique, nous demandons au décideur de comparer des alternatives « similaires » dont le but est de pouvoir mieux ajuster les paramètres du modèle SRMP de sorte qu'il puisse discriminer entre deux alternatives similaires. Deux alternatives sont similaires si la distance entre elles est petite. Dans ce cas particulier, nous utilisons la norme $L1$ $d_{(a,b)}$, mais d'autres types de distances peuvent être utilisées.

$$d_{(a,b)} = \frac{\sum_{j=1}^m |a_j - b_j|}{m}.$$

$$(a, b)_{\text{sim}} = \underset{(x,y) \in \mathcal{D}}{\text{argmin}} d_{(x,y)}$$

6.2.1.3 \mathcal{H}_{dis} : heuristique d'alternatives dissimilaires

En utilisant cette heuristique, on demande au décideur de comparer des alternatives « dissimilaires », en utilisant à nouveau dans ce cas la norme $L1$.

$$(a, b)_{\text{dis}} = \underset{(x,y) \in \mathcal{D}}{\text{argmax}} d_{(x,y)}$$

6.2.2 Heuristiques utilisant l'ensemble des solutions

Ces heuristiques utilisent le modèle SRMP actuellement construit (M_i) afin de sélectionner la paire d'alternatives suivante. Autrement dit, les préférences précédentes exprimées par le décideur sont prises en considération dans le processus de choix de la prochaine paire. En nous basant sur ce concept, nous proposons de différentes heuristiques dans la suite.

6.2.2.1 \mathcal{H}_{cp} : heuristique d'alternatives proches d'un profil

Cette heuristique permet de sélectionner à chaque itération la paire d'alternatives la plus proche d'un profil. Nous espérons que cette paire va permettre d'ajuster au mieux ce profil.

Lors de la première itération, puisque nous n'avons pas de modèle SRMP actuel, nous fixons un nombre de paires (par exemple 10% du nombre total de paires) qui est choisi au hasard parmi \mathcal{D} pour construire un premier modèle SRMP M_0 . Ensuite, nous sélectionnons itérativement la paire la plus proche du profil. La sélection d'une telle paire est détaillée ci-dessous.

Le concept de l'utilisation d'une distance est également maintenu ici, mais utilisé dans un contexte différent. Puisque nous utilisons le modèle actuel M_i , nous utiliserons ses paramètres pour sélectionner la prochaine paire d'alternatives. Pour refléter ceci, la distance entre l'alternative $a \in \mathcal{A}$ et un profil $p \in \{p^1, p^2, \dots, p^k\}$ est une distance pondérée définie comme suit :

$$d_{(a,p)} = \frac{\sum_{j=1}^m w_j |a_j - p_j|}{m},$$

où w_j correspondent aux poids des critères du modèle M_i .

Puisqu'il s'agit d'une paire d'alternatives de \mathcal{D} , nous considérons ici la distance moyenne $d_{((a,b),p)}$ entre $d_{(a,p)}$ et $d_{(b,p)}$:

$$d_{((a,b),p)} = \frac{(d_{(a,p)} + d_{(b,p)})}{2}$$

L'heuristique \mathcal{H}_{cp} sélectionne la paire d'alternatives la plus proche d'un profil donné p :

$$(a, b)_{\text{cp}} = \operatorname{argmin}_{(x,y) \in \mathcal{D}} d_{((x,y),p)}$$

Le profil utilisé par cette heuristique est choisi de manière séquentielle.

6.2.2.2 \mathcal{H}_{fp} : heuristique d'alternatives loin d'un profil

Comme pour l'heuristique précédente, nous définissons également une heuristique qui sélectionne la paire d'alternatives la plus éloignée d'un profil p donné :

$$(a, b)_{\text{fp}} = \operatorname{argmax}_{(x,y) \in \mathcal{D}} d_{((x,y),p)}$$

6.2.2.3 \mathcal{H}_{mp} : heuristique d'alternatives maximisant le nombre de profils nécessaires pour les discriminer

Pour cette heuristique, la première itération est une sélection aléatoire d'une paire d'alternatives de \mathcal{D} . Ensuite, à chaque itération, nous utiliserons le modèle généré dans l'itération précédente M_i pour sélectionner la paire d'alternatives suivante. L'idée est de sélectionner une paire qui, dans le modèle actuel M_i , utilise le plus grand nombre possible de profils dans sa comparaison afin de les distinguer (idéalement une paire considérée comme indifférente par M_i).

En confrontant le décideur à une telle paire, nous espérons que sa réponse générera une nouvelle contrainte pour le MIP qui réduira significativement la taille de l'espace de recherche.

Nous commençons par sélectionner une paire (a, b) qui est indifférente en utilisant M_i . Cela signifie que pour (a, b) tous les profils k ont été testés par la procédure d'exécution du modèle SRMP, et le modèle M_i n'a pas pu dire si a est préféré à b ou b est préféré à a . S'il n'y a pas de paire indifférente, une paire qui utilise tous les profils k dans M_i afin de discriminer a et b de cette manière sera sélectionnée. S'il n'en existe pas, nous recherchons une paire à l'aide des profils $k - 1$, jusqu'à atteindre le cas d'un profil.

Pour modéliser cela de manière plus formelle, associons un score de pénalité qui est un nombre entier avec les propositions logiques du tableau 6.4. Avec chaque paire $(a, b) \in \mathcal{D}$ nous associons la pénalité de la première proposition qui est vraie dans la séquence du tableau 6.4. Enfin, l'heuristique \mathcal{H}_{mp} sélectionne une paire (a, b) parmi \mathcal{D} qui minimise cette pénalité.

Pénalité	Proposition
1	$(a \sim_{p^{\sigma(1)}} b \text{ and } \dots \text{ and } a \sim_{p^{\sigma(k)}} b)$
2	$(a \sim_{p^{\sigma(1)}} b \text{ and } \dots \text{ and } a \sim_{p^{\sigma(k-1)}} b \text{ and } a \succ_{p^{\sigma(k)}} b)$
...	...
k	$(a \sim_{p^{\sigma(1)}} b \text{ and } a \succ_{p^{\sigma(2)}} b)$
k+1	$(a \succ_{p^{\sigma(1)}} b)$

Tableau 6.4 – Fonction de pénalité de \mathcal{H}_{mp}

6.3 Configuration du MIP

Afin d'obtenir dans la phase d'élicitation incrémentale des préférences un modèle satisfaisant le plus rapidement possible, c.-à-d. avec le moins d'itérations possibles, nous abordons également la question du choix d'un modèle à chaque itération, parmi tous les modèles compatibles. Puisque nous utilisons un programme linéaire mixte en nombres entiers, tel que celui présenté en section 5.1.1, cette question se traduit par la formulation de sa fonction objectif. Encore une fois, comme auparavant, certaines de ces fonctions objectifs prennent en compte les modèles générés dans les itérations précédentes, ce qui est présenté par une flèche en pointillé dans la figure 6.4, alors que d'autres ne le font pas.

Définissons d'abord une distance $d_{(M_i, M_j)}$ entre deux modèles SRMP M_i et M_j basée sur la norme $L1$.

$$d_{(M_i, M_j)} = \alpha * d_{(P_{M_i}, P_{M_j})} + \beta * d_{(W_{M_i}, W_{M_j})}, \text{ où :}$$

$$d_{(P_{M_i}, P_{M_j})} = \sum_{h=1}^k \frac{\sum_{l=1}^m |p_{(M_i, l)}^h - p_{(M_j, l)}^h|}{m}$$

$$d_{(W_{M_i}, W_{M_j})} = \frac{\sum_{l=1}^m |w_{(M_i, l)} - w_{(M_j, l)}|}{m}$$

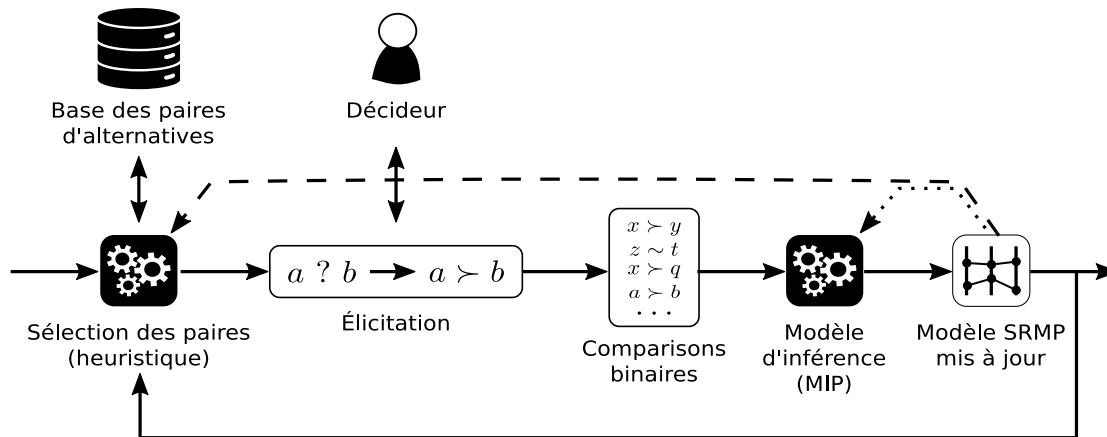


Figure 6.4 – Processus d'élitication incrémentale

$p_{(M_i, l)}^h$ est la performance du profil p^h sur le critère l dans le modèle M_i , tandis que $w_{(M_i, l)}$ est le poids du critère l dans le modèle M_i . Les coefficients α et β (avec $\alpha, \beta \geq 0$ et $\alpha + \beta = 1$) permettent de pondérer l'importance relative des profils par rapport aux poids dans le calcul des distances. Cette distance représente la ressemblance des deux modèles SRMP en termes de proximité des paramètres (profils et poids) d'un modèle à l'autre.

6.3.1 MIP_{first} : premier modèle SRMP réalisable

Dans cette configuration, nous utilisons la première solution réalisable donnée par le solveur (dans nos expériences IBM Cplex [IBM ILOG, 2018]) donc la fonction objectif est simplement une fonction constante :

minimize constant

6.3.2 MIP_{close} : modèle SRMP le plus proche du modèle précédent

Pour cette configuration du MIP, nous recherchons le modèle SRMP le plus proche du modèle précédent en termes de la distance définie précédemment. Cela devrait permettre de stabiliser en quelque sorte le modèle à chaque itération, et d'éviter qu'il ne change trop radicalement. De nouveaux paramètres ainsi que de nouvelles variables et contraintes sont ajoutés au MIP initial (tableau 5.1).

La nouvelle version de ce MIP est présentée dans le tableau B.1 en annexe. Le modèle précédent est ajouté comme paramètre à ce MIP dans (3) et (4), les coefficients α et β sont ajoutés aussi dans (5) et (7). Les (14), (15), (16) et (17) sont des variables utilisées pour le calcul de la distance. Ce calcul génère quatre contraintes supplémentaires liées à l'utilisation des valeurs absolues (36), (37), (38) et (39).

La fonction objectif de cette configuration est de minimiser la distance entre le modèle SRMP précédent et le modèle SRMP actuel et elle est présentée dans le MIP (tableau B.1) dans (18) :

$$\text{minimize } d_{(M_i, M_j)}$$

6.3.3 MIP_{center} : un modèle SRMP placé au centre de l'espace de recherche

Cette configuration vise à trouver le modèle SRMP le plus « général » parmi l'ensemble des solutions possibles. Par « général », nous entendons ici un modèle plus ou moins centré dans l'espace de solution afin d'éviter les solutions extrêmes. Pour avoir un tel modèle, nous décidons d'abord de trouver deux modèles extrêmes dans l'ensemble des solutions, c'est-à-dire les deux modèles les plus éloignés en termes de distance définie précédemment. En d'autres termes, un premier programme linéaire mixte en nombres entiers vise à trouver deux modèles SRMP réalisables M_{min} et M_{max} simultanément dans le but de maximiser la distance entre ces modèles :

$$\text{maximize } d_{(M_{min}, M_{max})}$$

Ce MIP est détaillé dans tableau B.2. Puisque nous utilisons la distance entre deux modèles comme définie précédemment, les coefficients α et β sont rajoutés dans (3) et (5) aux paramètres du MIP. La liste des variables liées à un modèle SRMP (critères et profils) (7), (8), (9), (10) et (11). Toutes les contraintes utilisées dans le MIP initial (tableau 5.1) sont également dédoublées puisque nous cherchons deux modèles qui respectent la liste des comparaisons par paires donnée comme paramètre.

Ces deux modèles extrêmes seront utilisés pour trouver le modèle central M_C que nous recherchons. Nous utilisons un deuxième programme linéaire mixte en nombres entiers présenté dans le tableau B.3. Il prend comme paramètres en plus les deux modèles M_{min} et M_{max} ((3), (4), (5), (6) et (11)) avec une fonction objectif qui minimise la distance entre M_C et M_{min} et la distance entre M_C et M_{max} . Comme cela peut mener à plusieurs modèles centraux possibles, nous minimisons également la différence entre ces deux distances pour garantir que M_C sera le plus proche de M_{max} et M_{min} en même temps. Cela est représenté par une variable $\Delta_{min,max}$ ((24)). Donc la fonction objective est de la forme ((26)) :

$$\text{minimize } (d_{(M_C, M_{min})} + d_{(M_C, M_{max})} + \Delta_{min,max})$$

Quatre contraintes supplémentaires (43), (44), (45) et (46), sont utilisées pour le calcul de $\Delta_{min,max}$.

6.4 Validation empirique

Notre objectif est de réduire l'effort cognitif du décideur en lui posant le minimum de nombre de questions. Le but de cette section est d'étudier empiriquement quelle combinaison de configuration heuristique / MIP conduit au meilleur modèle SRMP avec le moins de comparaisons par paires possibles.

6.4.1 Plan d'expériences

Nos expériences suivent le processus d'élicitation incrémentale présenté dans la section 6.1.2 avec une phase supplémentaire pour tester la qualité du modèle SRMP obtenu et son comportement face à de nouvelles alternatives. Une base de données \mathcal{D} de 100 paires d'alternatives est utilisée comme entrée pour toutes les heuristiques proposées de la section 6.2, combinée avec les 3 configurations MIP possibles de la section 6.3. L'heuristique choisie sélectionne une paire d'alternatives parmi \mathcal{D} à chaque itération. Le décideur de la figure 6.2 est remplacé pour nos expériences par un modèle SRMP généré aléatoirement M_{DM} , qui est utilisé pour comparer des paires d'alternatives, ce qui génère de nouvelles contraintes pour le modèle M_i . Nous considérons ici qu'il n'y a pas d'incohérences, que les réponses du décideur sont toujours compatibles avec les précédentes et avec un modèle SRMP.

Pour étudier le comportement d'un modèle, nous avons également une base de données de test D_{test} composée de 5000 alternatives générées aléatoirement. Ces alternatives sont classées à la fois par le modèle SRMP original M_{DM} et le modèle courant M_i . La qualité de M_i est ensuite évaluée en utilisant la mesure de corrélation de rang τ de *Kendall* entre ces deux classements. Pour rappel, nous notons $kPmC$ les instances de problèmes générés avec k profils et m critères.

Nous exécutons ce processus pour 100 bases de données artificielles différentes \mathcal{D} , composées chacune de 100 paires d'alternatives, évaluées sur différents nombres de critères ($m = 3, 5, 7$). Nous fixons le nombre de profils du modèle SRMP à 2.

Les calculs sont effectués sur plusieurs serveurs configurés avec 20 cœurs de calcul (ce qui permet à CPLEX d'atteindre un parallélisme de 20 pour la résolution du MIP) et 30 Go de mémoire vive.

6.4.2 Résultats expérimentaux et discussion

La figure 6.5 représente la valeur moyenne du tau de *Kendall* pour chaque itération de l'élicitation incrémentale (une paire supplémentaire d'alternatives sélectionnées à chaque itération) pour la configuration $\mathcal{H}_{rnd} \setminus MIP_{first}$ pour un problème de 2 profils et 3 critères ($2P\ 3C$). Nous observons que la valeur du tau de *Kendall* augmente et tend vers 1 avec le nombre de paires d'alternatives utilisées pour apprendre le modèle SRMP. Cela signifie que plus nous utilisons de paires d'apprentissage, plus M_i est proche de M_{DM} . L'écart type associé aux valeurs moyennes illustrées dans cette figure est faible et il diminue également avec l'ajout de paires d'apprentissage supplémentaires et il est en moyenne 0.095.

La figure 6.6 représente la valeur moyenne du tau de *Kendall* pour les différentes heuristiques ($\mathcal{H}_{rnd}, \mathcal{H}_{sim}, \mathcal{H}_{dis}, \mathcal{H}_{cp}, \mathcal{H}_{mp}$) pour MIP_{first} et pour les problèmes $2P\ 3C$. Nous remarquons que les courbes des différentes heuristiques ont un comportement similaire où le tau de *Kendall* tend vers 1. Nous observons que l'heuristique \mathcal{H}_{mp} (courbe en bleu) domine les autres heuristiques. Pour valider cette observation, nous avons utilisé le test statistique *Kolmogorov-Smirnov* [Dodge, 2008]. Ce test permet de comparer deux échantillons avec des fonctions de répartitions inconnues en utilisant leurs fonction de répartitions empiriques.

Nous remarquons aussi que pour les premières itérations les courbes ont un comportement similaire ce qui est dû au petit nombre de paires d'alternatives impliquées, et qui conduisent à des modèles SRMP pas très expressifs. Ensuite, les courbes se séparent clairement en faveur de \mathcal{H}_{mp} . Pour les dernières itérations, les courbes redeviennent similaires, ce qui s'explique par le fait que

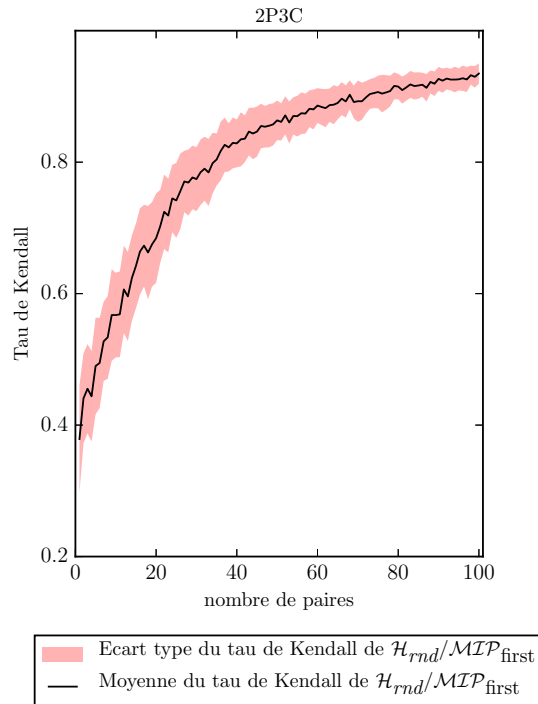


Figure 6.5 – Moyenne et écart type du tau de *Kendall* de l’heuristique \mathcal{H}_{rnd} pour les problèmes $2P3C$

l’ensemble des paires d’apprentissage est presque identique, indépendamment de l’heuristique de sélection (\mathcal{D} est fini et fixé à 100 paires d’alternatives).

La figure 6.7 représente la valeur moyenne du tau de *Kendall* des différentes heuristiques pour les trois configurations possibles du MIP ($\mathcal{MIP}_{\text{first}}$, $\mathcal{MIP}_{\text{close}}$ et $\mathcal{MIP}_{\text{center}}$), et pour deux tailles différentes des problèmes : 2 profils 3 critères ($2P3C$) et 2 profils 5 critères ($2P5C$). Comme nous l’avons remarqué avant, les écarts-types associés aux valeurs moyennes sont faibles et ils diminuent également avec l’ajout de paires d’apprentissage supplémentaires. Pour des raisons de clarté d’affichage, les écarts-types ne sont pas illustrés dans la figure, 6.7 mais par exemple nous avons en moyenne 0.095 pour les problèmes $2P5C$ pour \mathcal{H}_{mp} et \mathcal{H}_{rnd} .

Les résultats montrent une tendance attendue dans laquelle les valeurs du tau de *Kendall* augmentent avec le nombre de paires d’alternatives utilisées pour apprendre le modèle SRMP. Cela signifie que plus on utilise de paires d’apprentissage, plus M_i est proche de M_{DM} . Nous observons pour les différentes expériences présentées dans la figure 6.7 que le \mathcal{H}_{mp} (représenté en bleu) domine les autres. Pour valider cette observation, nous avons utilisé le test statistique *Kolmogorov-Smirnov*, qui permet de comparer deux échantillons [Dodge, 2008]. Par conséquent, nous comparons par paires les différentes courbes du tau de *Kendall* pour chacune des configurations et tailles de données, et nous confirmons que les courbes \mathcal{H}_{mp} dominent les autres pour les différentes configurations MIP et tailles de données.

Nous observons également que \mathcal{H}_{mp} atteint une valeur tau de *Kendall* plus rapidement

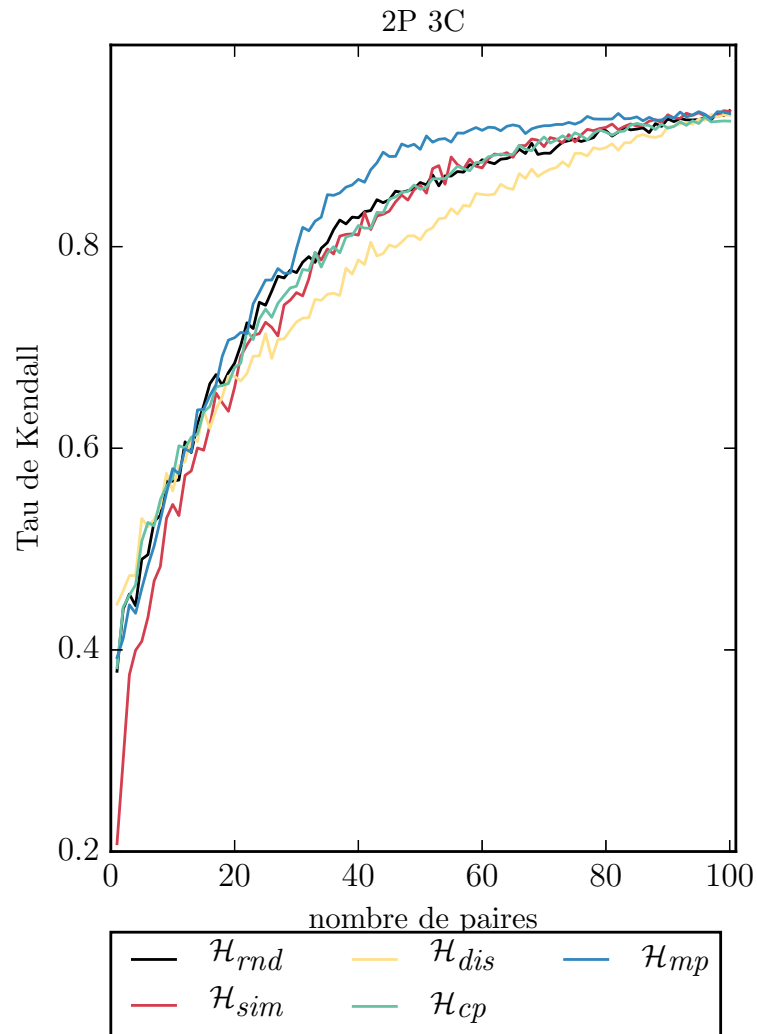
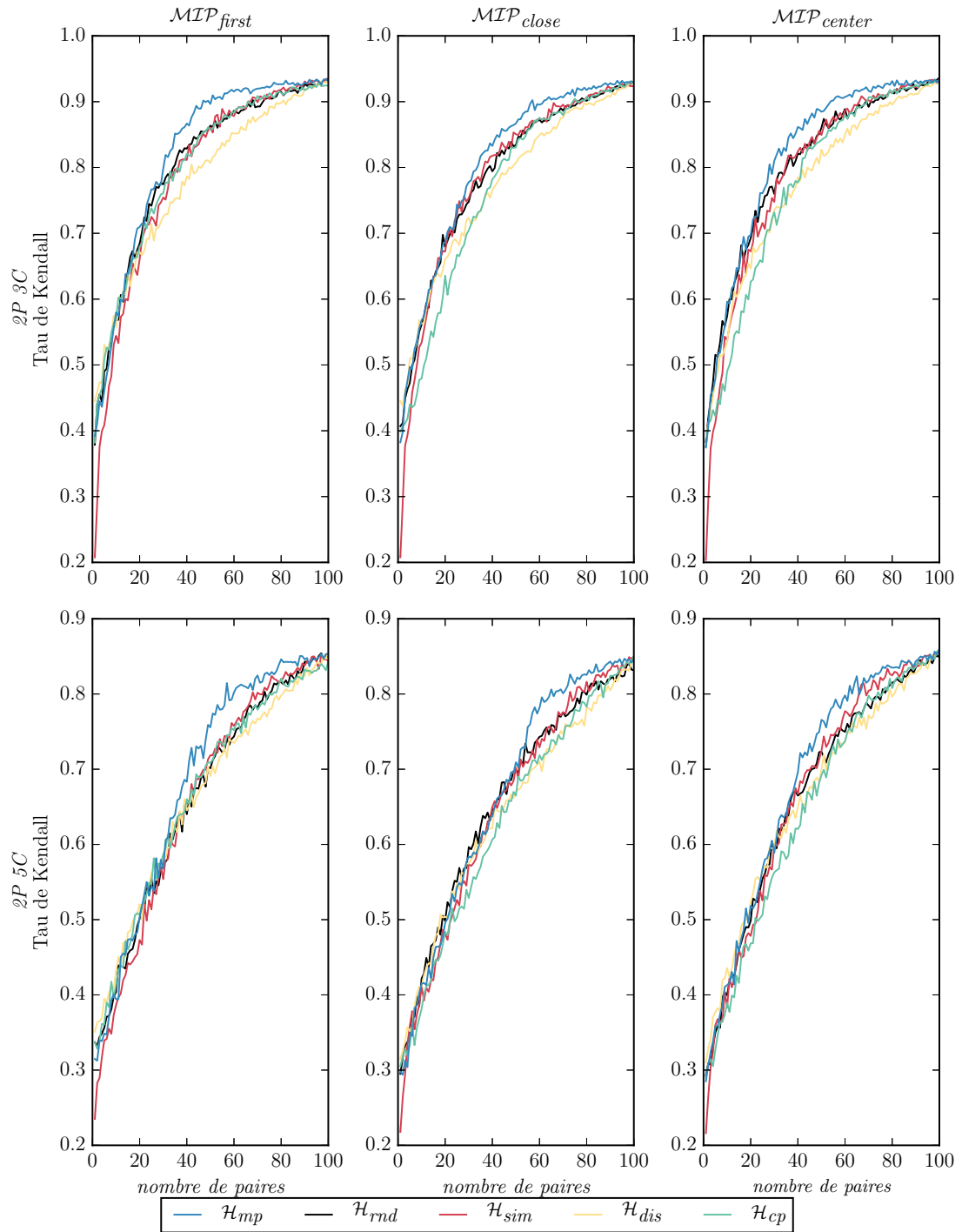


Figure 6.6 – Moyenne du tau *Kendall* des heuristiques pour $\mathcal{MIP}_{\text{first}}$ pour les problèmes $2P$ $3C$

que les autres (c.-à-d. avec moins d'exemples d'apprentissage). Par exemple pour le premier graphique de la figure 6.7 ($2P$ $3C$ / $\mathcal{MIP}_{\text{first}}$), \mathcal{H}_{mp} atteint approximativement un tau de *Kendall* de 0.9 avec environ 50 paires d'apprentissage, tandis que les autres heuristiques ont besoin de plus de 80 paires pour atteindre la même valeur du tau de *Kendall*.

Ces observations nous aident à choisir l'heuristique \mathcal{H}_{mp} parmi toutes celles proposées, mais pas la configuration MIP ($\mathcal{MIP}_{\text{first}}$, $\mathcal{MIP}_{\text{close}}$ et $\mathcal{MIP}_{\text{center}}$). Comme le montre la figure 6.8 il n'est pas si facile de choisir spécialement entre les configurations $\mathcal{MIP}_{\text{first}}$ et $\mathcal{MIP}_{\text{center}}$. Pour cela, nous avons décidé d'examiner le temps de calcul qui inclut la sélection des paires d'apprentissage en utilisant les heuristiques proposées et la résolution du MIP.

Figure 6.7 – Moyenne du tau *Kendall* pour les problèmes $2P\ 3C$ et $2P\ 5C$

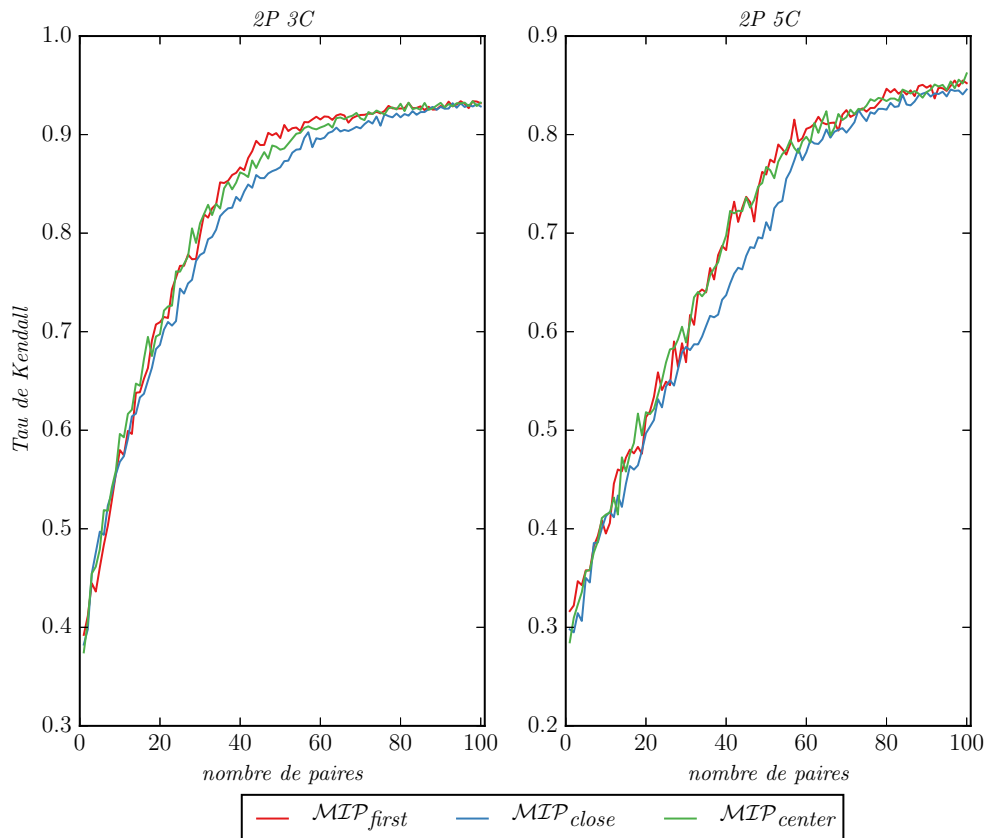
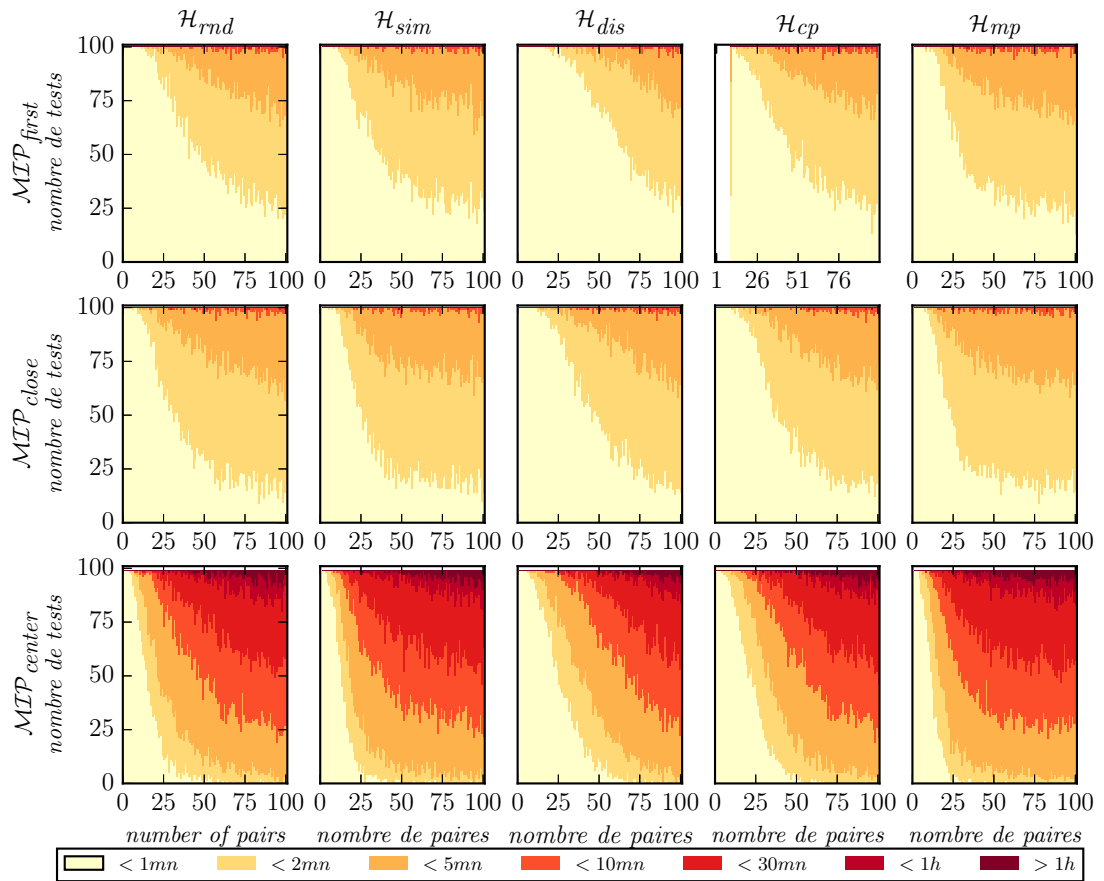
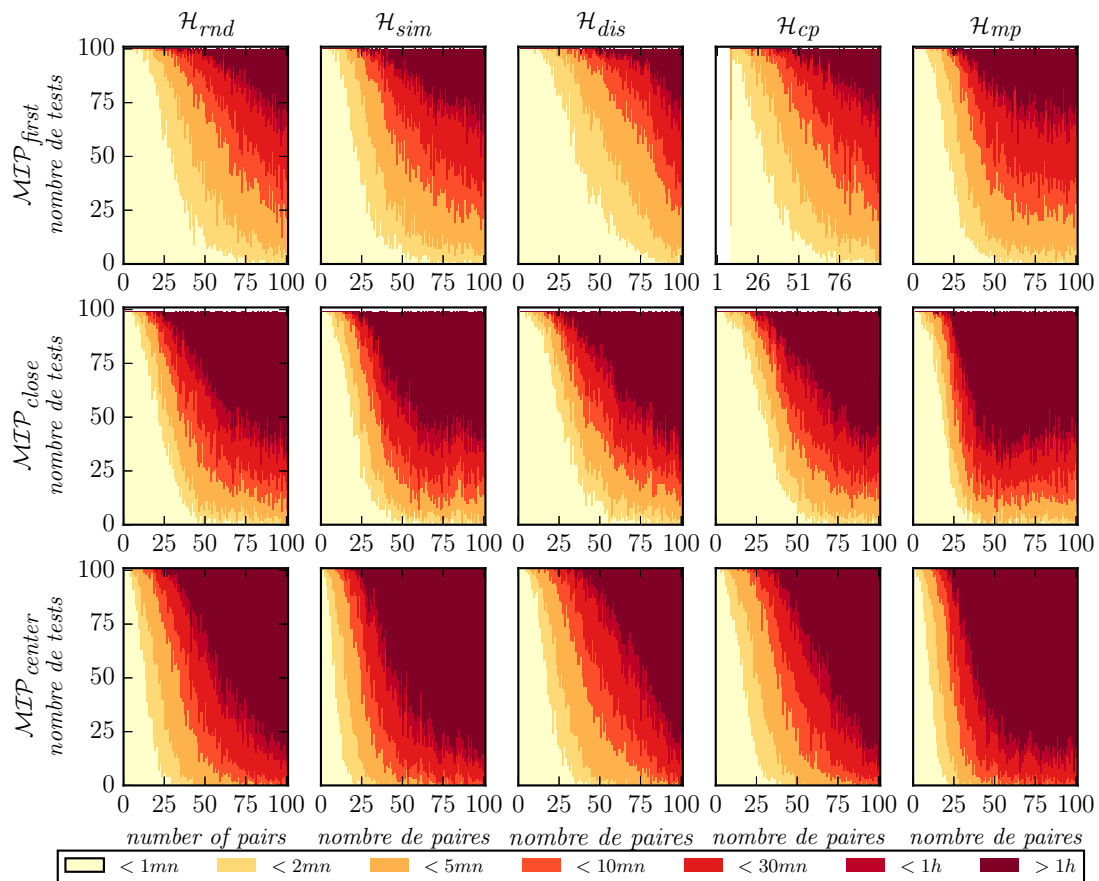


Figure 6.8 – Moyenne du tau de Kendall de \mathcal{H}_{mp} pour les problèmes $2P\ 3C$ et $2P\ 5C$

La figure 6.9 et la figure 6.10 représentent à chaque itération le nombre de tests ayant un temps d'exécution *inférieur à une, deux, cinq, dix et trente minutes, moins d'une heure et plus qu'une heure* pour chacune des combinaisons d'heuristique et du MIP pour respectivement $2P\ 3C$ et $2P\ 5C$. Par exemple dans la figure 6.9 pour $\mathcal{H}_{mp} / MIP_{first}$ nous pouvons voir que pour 30 cas sur 100 le temps de calcul pour apprendre un modèle SRMP avec 50 paires d'apprentissage est inférieur à 1 minute. Pour la même configuration du $2P\ 5C$ (figure 6.10) pour 20 cas sur 80, le temps de calcul est inférieur à 30 minutes.

Figure 6.9 – Temps d'exécution pour les problèmes $2P\ 3C$

Figure 6.10 – Temps d'exécution pour les problèmes $2P\ 5C$

Nous observons que plus le problème est complexe, plus le temps de calcul augmente et plus la configuration $\mathcal{MIP}_{\text{first}}$ prend beaucoup moins de temps que $\mathcal{MIP}_{\text{close}}$ et $\mathcal{MIP}_{\text{center}}$ (pour les deux dimensions du problème). Par conséquent, un bon compromis entre précision et temps de calcul nous amène à proposer la combinaison de configuration de l'heuristique et du MIP est $\mathcal{H}_{\text{mp}} / \mathcal{MIP}_{\text{first}}$.

Afin d'étendre notre étude, un problème plus complexe (c'est-à-dire 2 profils et 7 critères) est testé, mais comme indiqué dans le tableau 6.5 le temps nécessaire pour un test exhaustif des problèmes $2P\ 5C$ est environ un an avec plus de vingt instances de serveur avec $20\ CPU$ s et $30GB\ RAM$ chacune. De plus, la tendance observée dans la figure 6.9 et la figure 6.10 montre que le temps de calcul augmente considérablement avec le nombre de critères. Pour cette raison, pour les problèmes contenant 7 critères, nous avons décidé d'étudier uniquement la configuration $\mathcal{MIP}_{\text{first}}$ et de comparer la configuration \mathcal{H}_{mp} heuristique avec celle \mathcal{H}_{rnd} .

	Random	Similar	Dissimilar	CloseProfile	MaxProfiles
First feasible	223	331	135d	190	450
Min Distance	616	689	600	517	828
Center	793	1001	556d	659	1151

Tableau 6.5 – Temps d'exécution global approximatif en jours pour un test exhaustif des problèmes $2P\ 5C$

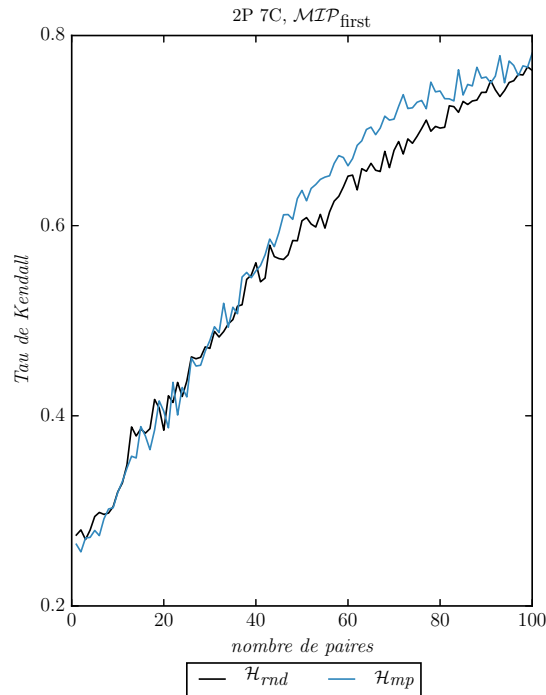


Figure 6.11 – Moyenne du tau de *Kendall* de \mathcal{H}_{mp} et \mathcal{H}_{rnd} pour les problèmes $2P\ 7C$

La figure 6.11 représente la moyenne du tau de *Kendall* à chaque itération de \mathcal{H}_{mp} et \mathcal{H}_{rnd} pour les problèmes *2P 7C*. Nous confirmons notre observation précédente que \mathcal{H}_{mp} domine \mathcal{H}_{rnd} en utilisant le test *Kolmogorov-Smirnov*.

6.4.3 En pratique

Dans cette section, nous présentons comment nos résultats peuvent être utilisés dans un contexte pratique et donnons une réponse à l'une de nos questions de recherche qui est de savoir combien de paires d'apprentissages / itérations sont nécessaires pour atteindre un modèle SRMP « satisfaisant » et arrêter le processus d'élicitation incrémentale.

Un modèle SRMP est considéré comme « satisfaisant » par le *décideur* si ce modèle peut atteindre une valeur donnée du tau de *Kendall*. Une fois que celle-ci est donnée, nous utilisons les courbes de la moyenne du tau de *Kendall* pour trouver le nombre de paires nécessaires pour atteindre cette valeur. Par exemple, la figure 6.12 représente la moyenne du tau de *Kendall* de $\mathcal{H}_{\text{mp}} / \mathcal{MIP}_{\text{first}}$ sur la gauche, et le temps cumulé de l'exécution à droite pour *2P 3C*, où le *décideur* fixe $\tau = 0,9$. Nous observons que nous avons besoin d'environ 48 paires d'apprentissage et que cela prendrait environ 42 minutes au total pour les calculs avec notre configuration de serveur (à gauche de la figure 6.12). Pour un problème *2P 5C* le *décideur* fixe $\tau = 0,8$ où environ 56 paires d'apprentissage (à gauche de la figure 6.13) sont requises et peuvent être exécutées en 1600 minutes environ (à droite de la figure 6.13).

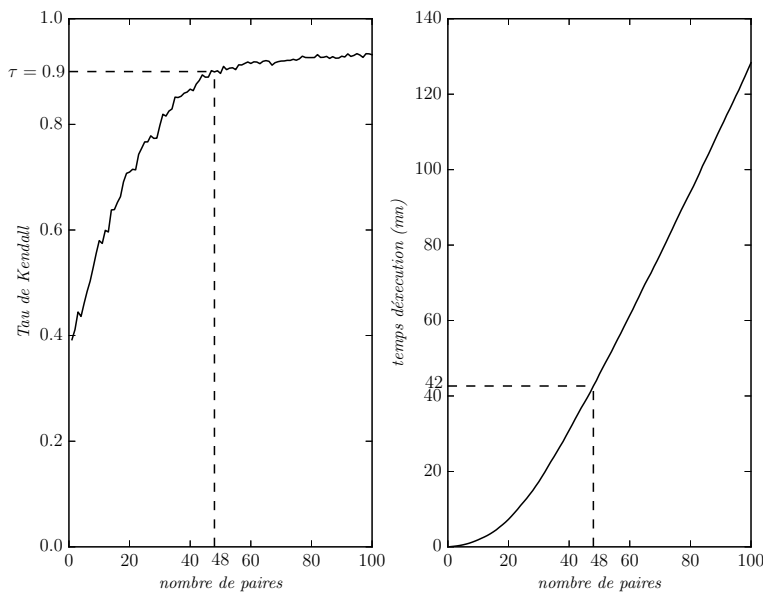


Figure 6.12 – Tau de *Kendall* moyen et temps d'exécution moyen cumulé de $\mathcal{H}_{\text{mp}} / \mathcal{MIP}_{\text{first}}$ pour les problèmes *2P 3C*

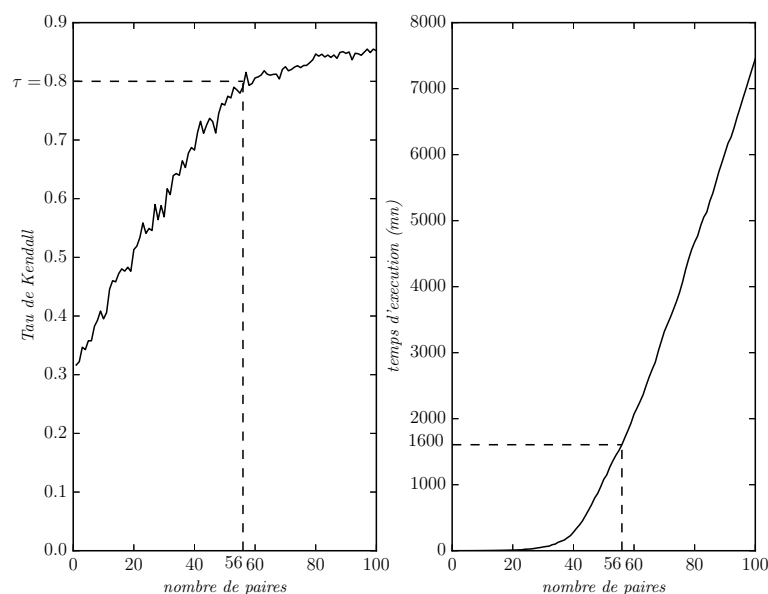


Figure 6.13 – Tau de *Kendall* moyen et temps d’exécution moyen cumulé de $\mathcal{H}_{mp} / \mathcal{MIP}_{\text{first}}$ pour les problèmes *2P 5C*

Cette partie de validation empirique nous a permis d’étudier les différentes heuristiques que nous avons proposées et de les tester avec les différentes configurations du MIP. Cela nous a permis de conclure que la meilleure configuration d’heuristique et de MIP est $\mathcal{MIP}_{\text{first}} \setminus \mathcal{H}_{mp}$. D’autre part cela nous a permis de trouver la mesure d’arrêt du processus d’élitication incrémentale qui dépend de la satisfaction du décideur sur le modèle. Mais nous sommes bien conscients du problème du temps d’exécution. Pour cette raison nous avons décidé de remplacer le programme linéaire mixte en nombres entiers par la métaheuristique présentée dans la section 5.2. Nous montrons notamment que la qualité des solutions sera inférieure à celles obtenues par la programmation mathématique, mais que les gains en temps de calcul l’emportent largement, notamment à cause du caractère incrémental et interactif du processus que nous proposons.

6.5 Élicitation incrémentale avec l’algorithme génétique

L’utilisation d’un programme linéaire mixte en nombres entiers pour trouver la solution exacte pour inférer le modèle SRMP à chaque itération du processus d’élitication incrémentale provoque un problème de temps d’exécution. Cela est un handicap pour l’utiliser dans un cas réel avec un vrai décideur humain et des problèmes de plus grande taille. Pour cette raison nous proposons d’utiliser notre algorithme génétique (algorithme 4) présenté dans le chapitre 5 dans le processus d’élitication incrémentale (figure 6.4) à la place du MIP.

Pour montrer l’apport d’utiliser notre algorithme génétique dans l’élitication incrémentale, nous étudions empiriquement l’heuristique de sélection de paires d’apprentissage choisies dans la section précédente par la méthode exacte, \mathcal{H}_{mp} . Dans cette étude nous utilisons les mêmes données que celles utilisées pour le MIP pour pouvoir comparer les résultats issus des deux études.

Nous suivons le processus d'élicitation incrémentale (section 6.1.2) pour nos expériences. L'heuristique \mathcal{H}_{mp} sélectionne une paire d'alternatives dans une base de 100 paires \mathcal{D} à chaque itération. Le décideur est remplacé par un modèle SRMP généré aléatoirement M_{DM} qui permet de comparer deux alternatives. Ensuite cette comparaison est rajoutée à liste des comparaisons par paires pour inférer un nouveau modèle SRMP M_i . Une base de données de 5000 alternatives générée aléatoirement est utilisée pour mesurer la qualité du modèle M_i . Cette mesure est obtenue en calculant le tau de *Kendall* τ entre le classement des 5000 alternatives de \mathcal{D} par les deux modèles SRMP M_{DM} et M_i .

Ce processus est exécuté pour 100 bases de données artificielles différentes \mathcal{D} . Chaque base est composée de 100 paires d'alternatives évaluées sur différents nombres de critères ($m = 3, 5, 7$). La complexité des modèles SRMP varie entre 2 et 3 profils.

La figure 6.14 montre la moyenne du tau de *Kendall* pour chaque itération en utilisant \mathcal{H}_{mp} comme stratégie de sélection des paires et notre algorithme génétique pour inférer le modèle SRMP M_i pour les problèmes de 3 critères et 2 profils $2P\ 3C$. Nous observons que la valeur moyenne du tau de *Kendall* augmente à chaque itération où une paire d'apprentissage est ajoutée. Donc le modèle M_i se rapproche plus du modèle original M_{DM} en augmentant le nombre de paires d'apprentissage. L'écart type montré dans la figure (en rose) est faible et diminue avec l'ajout des paires d'alternatives.

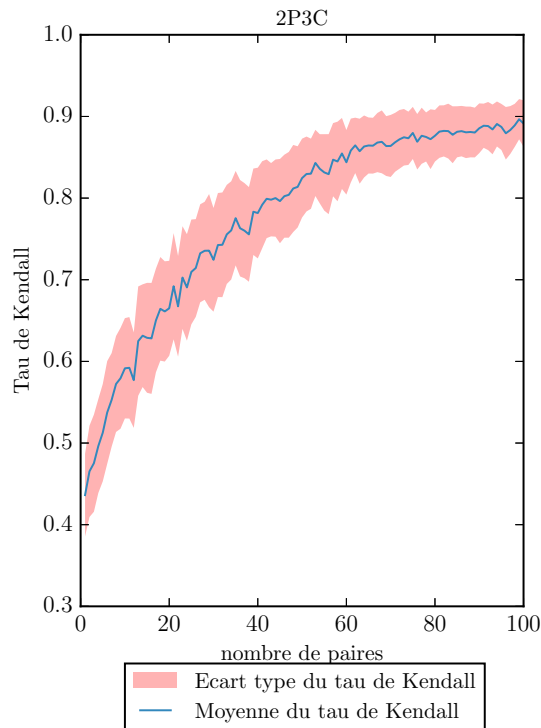


Figure 6.14 – Moyenne et écart type du tau de *Kendall* avec l'algorithme génétique pour les problèmes $2P3C$

L'utilisation de l'algorithme génétique garde bien la tendance où la valeur moyenne du tau de *Kendall* augmente avec le nombre de paires d'apprentissage pour les différents problèmes : $2P\ 3C$, $2P\ 5C$, $2P\ 7C$, $3P\ 3C$, $3P\ 5C$ et $3P\ 7C$ (comme montré dans la figure 6.15). Nous remarquons que la valeur moyenne maximale de τ atteinte diminue avec l'augmentation du nombre de critères. Par exemple pour les problèmes $2P\ 3C$ avec 100 paires d'alternatives la valeur moyenne de τ est de 0.89 alors que pour les problèmes $2P\ 7C$ est de 0.75. Nous concluons ici que plus le problème est complexe plus le nombre nécessaire de paires d'apprentissage pour atteindre un tau de *Kendall* assez élevé augmente.

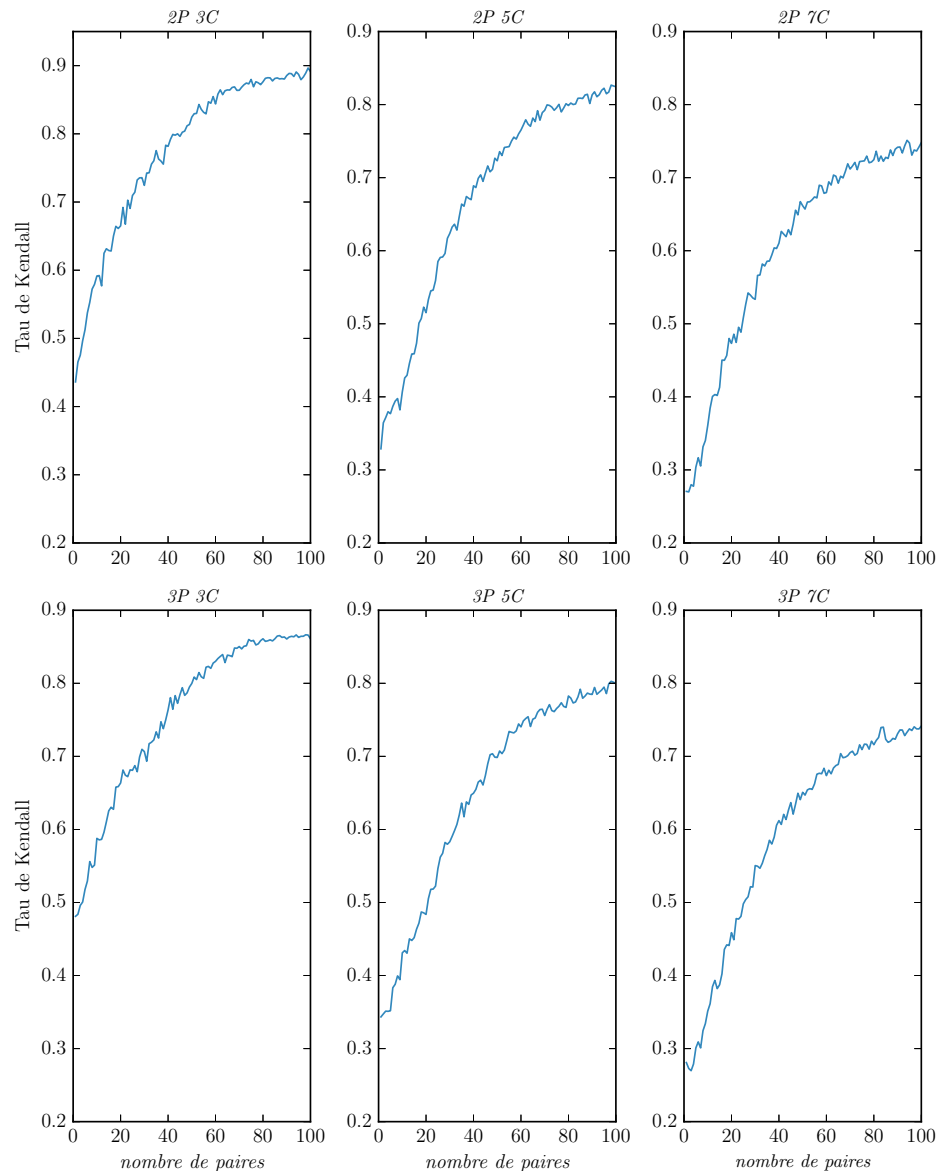


Figure 6.15 – Moyenne du tau de *Kendall* avec l'algorithme génétique pour les différents problèmes

La figure 6.16 montre le temps de calcul moyen et l'écart type pour chaque itération pour les problèmes $2P\ 3C$ et $3P\ 7C$ (attention au changement d'échelle entre les deux graphiques, pour $2P\ 3C$ il s'agit de millisecondes, alors que pour $3P\ 7C$ l'échelle des ordonnées est en secondes). Nous remarquons que pour les problèmes les moins complexes ($2P\ 3C$) les temps de calcul se situent sous une demi-seconde (pour chaque itération), alors que pour les plus complexes de notre étude ($3P\ 7C$), ce temps reste sous 5s pour chaque itération. Ceci se confirme avec le reste des résultats qui sont présentés dans l'annexe C.

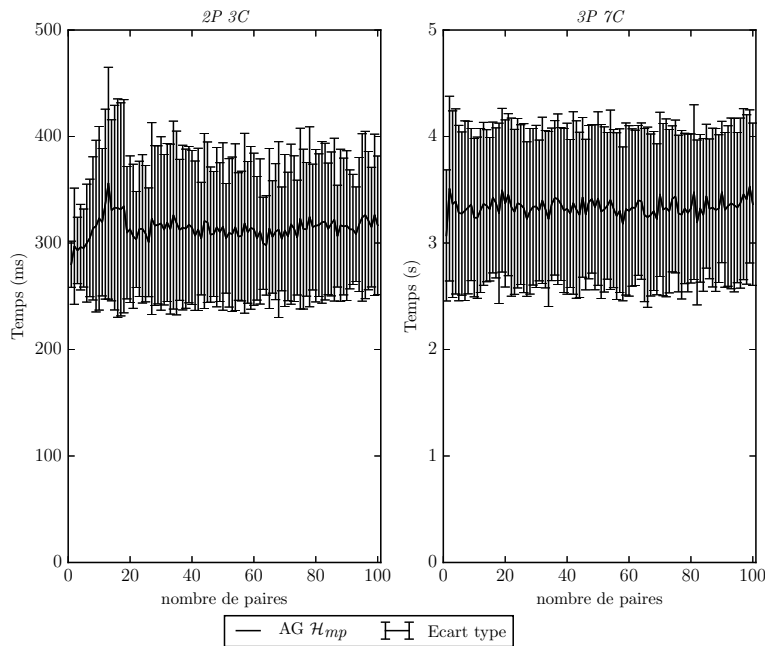


Figure 6.16 – Le temps d'exécution moyen et l'écart type avec l'algorithme génétique (AG) pour les problèmes $2P\ 3C$ et $3P\ 7C$

Afin de comparer les performances des deux méthodes exacte (MIP) et approchée (notre algorithme génétique), nous présentons dans la figure 6.17 la valeur moyenne du tau de *Kendall* pour les problèmes $2P\ 3C$ et $2P\ 5C$ obtenue par les deux méthodes. Nous remarquons que la courbe de la moyenne du tau de *Kendall* obtenue en utilisant le MIP (en rouge) domine celle obtenue par notre algorithme génétique (en bleue).

Dans le cas pratique pour atteindre une valeur de tau de *Kendall* de 0.8 pour un problème $2P\ 3C$, pour le MIP nous avons besoin d'environ 30 paires d'alternatives alors que pour notre algorithme il faut environ 50 paires. Pour atteindre un $\tau = 0.9$ il faut environ 60 paires d'apprentissage avec l'utilisation du MIP. Avec notre algorithme cette valeur n'est pas atteinte même avec toutes 100 paires d'alternatives ce qui montre la nécessité d'avoir plus de paires d'apprentissage pour l'atteindre.

Ces résultats se confirment pour des problèmes plus complexes, comme nous pouvons le constater dans la figure 6.15. Certains niveaux de tau de *Kendall* ne peuvent pas être atteints par la métaheuristique en moins de 100 itérations.

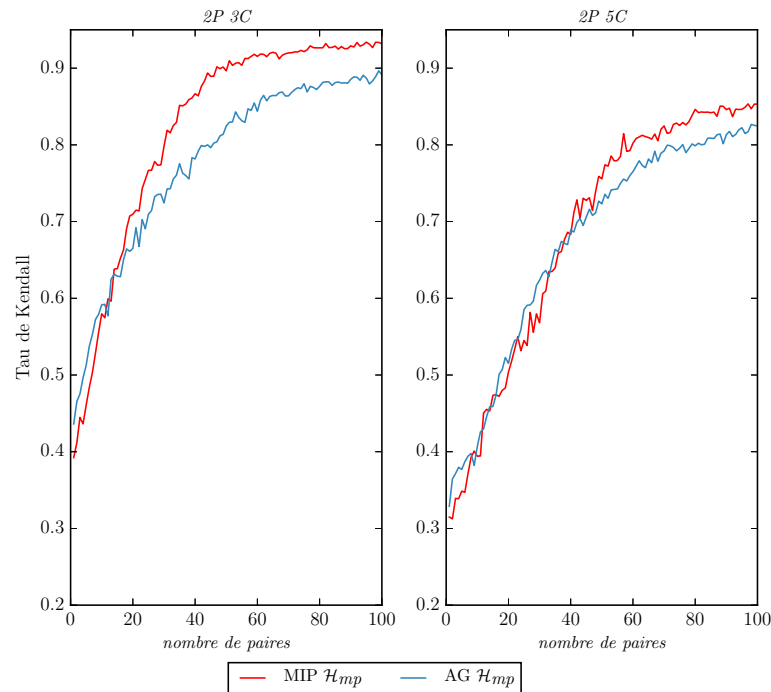


Figure 6.17 – Moyenne du tau de *Kendall* avec l'algorithme génétique (AG) et le MIP pour les problèmes *2P3C* et *2P5C*

Cependant, la métaheuristique présente un avantage non-négligeable sur le MIP dans notre processus incrémental. Les figures 6.9 et 6.10 montrent les temps de calcul moyen pour le MIP pour chaque itération. En les comparant à la figure 6.16, qui présente ces temps de calcul pour la métaheuristique, nous pouvons déduire que pour une élicitation incrémentale d'une centaine d'itérations, l'approche exacte nécessiterait plusieurs jours de calcul, alors que ce temps est de l'ordre de quelques minutes pour l'approche génétique.

6.6 Résumé et mise en perspective

Dans ce chapitre nous avons étudié l'élicitation incrémentale des modèles SRMP dont le but de réduire l'effort cognitif du décideur. Cette approche incrémentale est inspirée par le domaine de machine learning où elle est appelée l'apprentissage actif. Les données arrivent de manière séquentielle à un algorithme d'apprentissage qui utilise ces nouvelles données et le modèle précédemment appris ainsi que l'entrée d'un oracle pour produire un meilleur modèle. L'apprentissage actif se présente en général sous trois formes : la génération d'instances, l'apprentissage en ligne et hors ligne. Nous nous intéressons dans ce chapitre à l'apprentissage hors ligne et nous supposons qu'il existe une base de données d'instances et que l'algorithme sélectionne quelles données sont confrontées aux connaissances de l'oracle. Dans le contexte d'AMCD, cette approche est appelée élicitation incrémentale des préférences.

Des travaux sur l'élicitation incrémentale ont été déjà réalisés dans un cadre MAVT pour trouver une heuristique de sélection d'exemples d'apprentissage à chaque itération. Inspirés par ces travaux et le domaine de machine learning, nous étudions l'élicitation incrémentale des modèles SRMP pour réduire l'effort cognitif du décideur pendant ce processus.

Nous commençons par proposer un processus d'élicitation incrémentale des modèles SRMP. Une base de paire d'alternatives qui existe déjà est utilisée comme entrée pour une heuristique qui sélectionne une paire de la base. Ce choix d'une base de paires d'alternatives est justifié par notre contexte applicatif. En effet, dans le cas des décisions prises par un drone, il est important de comparer des alternatives dans un même contexte (c.-à-d. un waypoint par exemple). C'est pour cette raison que nous avons regroupé les alternatives par paires, afin de garantir une cohérence dans les comparaisons soumises au décideur. Le décideur est confronté à cette paire pour exprimer ses préférences (préférence stricte ou indifférence). Cette réponse génère une nouvelle comparaison binaire qui s'ajoute aux précédentes. Un modèle d'inférence utilise ces comparaisons pour construire un nouveau modèle des préférences. Ce processus interactif est répété jusqu'à ce que le décideur soit satisfait du modèle élicité, ou lorsqu'un critère d'arrêt est vérifié. Ce processus est conçu pour être générique c'est-à-dire que le modèle d'inférence peut être un algorithme d'élicitation des modèles SRMP à solutions exactes comme le MIP ou le SAT ou à solutions approchées comme la matheuristique ou notre algorithme génétique (chapitre 5).

Nous présentons aussi dans ce chapitre une façon de gérer l'apparition des informations imprécises pendant le processus d'élicitation incrémentale. Cela peut être lié à l'incertitude du décideur ou l'incompatibilité du modèle SRMP avec les préférences du décideur ou même à des données imprécises. Nous restreignons notre travail au premier scénario, où le décideur exprime ses préférences sur une nouvelle paire d'alternatives, mais qui sont incompatibles avec celles d'avant. Nous proposons une approche pour résoudre ces incompatibilités, en utilisant de la programmation en nombres entiers. Le résultat est sous la forme d'un ensemble d'ensembles de paires à enlever de la liste des comparaisons données par le décideur afin de rendre les réponses à nouveau compatibles avec un modèle SRMP.

Nous proposons et nous étudions, un ensemble d'heuristiques de sélection des paires d'alternatives afin d'en choisir une qui permet de réduire l'effort cognitif du décideur, autrement dit de réduire le nombre de paires d'alternatives nécessaires pour trouver un modèle qui représente au mieux ses préférences. Afin d'atteindre cet objectif nous proposons aussi différentes configurations du MIP qui sont combinées avec les heuristiques pour trouver la meilleure combinaison. Une validation empirique est réalisée pour trouver cette combinaison. Les résultats nous montrent que l'heuristique la plus efficace est celle qui permet d'interroger le décideur sur des paires considérées comme indifférentes dans le modèle courant. La configuration du MIP la plus efficace reste la première solution proposée par le solveur. Les autres configurations testées ont une complexité algorithmique plus élevée, et n'apportent que peu d'amélioration à la qualité du modèle des préférences courant.

Cette étude révèle aussi le problème lié au temps d'exécution très élevé, et par conséquent incompatible avec beaucoup d'applications réelles. Ceci nous motive à tester l'efficacité de notre algorithme génétique à la place du MIP dans l'inférence incrémentale. Les résultats montrent que les courbes des tau de *Kendall* sont dominées par celles du MIP et qu'il faut plus de paires d'apprentissage pour atteindre une valeur donnée du tau de *Kendall*. Cependant, ils montrent aussi que le temps de calcul devient négligeable par rapport à celui du MIP.

Le choix entre les deux approches, exactes ou approchées, est donc difficile et dépend principalement du contexte applicatif. Si le processus incrémental doit être mis en oeuvre en une fois dans une unique discussion avec le décideur, il est probablement mieux d'utiliser l'algorithme génétique, afin de garantir un bon niveau d'interactivité dans le processus incrémental. Le modèle des préférences trouvé ne sera par contre pas d'une qualité élevée. Par contre, si l'application permet une élicitation qui s'étale sur plusieurs discussions avec le décideur, il est possible d'utiliser le MIP dans le processus incrémental. Dans ce cas le modèle des préférences trouvé sera plus fidèle aux préférences réelles du décideur.

Conclusion et perspectives

Sommaire

Résumé	142
Discussion et limites	143
Perspectives	144

Les travaux menés durant ce travail doctoral se positionnent dans le contexte des drones autonomes. Un drone autonome est un aéronef qui ne transporte pas de pilote humain. Ce dernier est remplacé par un système qui lui permet de reproduire certaines de ses capacités sensorielles et décisionnelles. Dans le cadre de notre travail, le drone est considéré comme complètement autonome, c'est à dire qu'il est capable d'accomplir une mission sans l'intervention d'un opérateur humain et de prendre des décisions de façon totalement autonome lorsqu'il en a besoin. Il en découle la nécessité que l'opérateur au sol ait une confiance élevée dans les décisions prises par le drone.

Un premier objectif de cette thèse est donc de proposer un moteur de décisions à embarquer dans le drone autonome qui garantit un niveau élevé de confiance dans les décisions prises par le drone. Dans notre contexte, nous identifions trois facteurs sur lesquels nous pouvons influencer pour augmenter cette confiance de l'opérateur dans les décisions du drone. Tout d'abord, il est important que la structure et le fonctionnement du moteur de décisions soient faciles à expliquer à l'opérateur, pour éviter un effet de « boîte noire ». Ensuite, l'opérateur aura plus de confiance dans les décisions du drone si ce dernier se comporte selon ses attentes. Enfin, il est important de pouvoir interpréter et comprendre facilement les décisions prise par le drone et d'avoir la possibilité de retracer leur origine dans le moteur de décisions.

Pour contribuer à augmenter cette confiance, nous proposons également d'intégrer des modèles des préférences de l'opérateur dans le moteur de décisions, en utilisant des techniques d'Aide Multi-Critère à la Décision (AMCD). Ces modèles nécessitent cependant une phase d'élicitation des préférences de l'opérateur en amont de la mission, avant que cette configuration ne soit intégrée dans le drone. Pendant cette phase l'opérateur doit répondre à des questions qui permettent d'obtenir un modèle représentant au mieux ses préférences. Cela peut provoquer une fatigue cognitive chez l'opérateur, si ce nombre de questions est trop élevé. Un deuxième objectif de ce travail est donc de proposer un processus d'élicitation des préférences incrémentale dont le but est de minimiser l'effort cognitif de l'opérateur avant la mission.

Dans ce chapitre nous résumons d'abord nos contributions qui nous permettent de répondre à ces deux objectifs. Ensuite nous discutons des conséquences de ces contributions, tout en

insistant bien sur les limites de nos travaux. Ces limites nous permettent alors de présenter des perspectives pour nos travaux futurs.

Résumé

Dans ce travail nous proposons tout d'abord un moteur de décisions multi-niveaux inspiré par la logique adoptée par un pilote d'un aéronef lors d'une mission. Il est composé de deux niveaux de prise de décision de haut niveau. Le premier a pour but de décider si l'apparition d'un événement, qui est évalué sur plusieurs critères, nécessite un changement de plan de la mission ou non. Si c'est le cas, le deuxième niveau est déclenché. Ce dernier permet de choisir l'action de haut niveau la mieux adaptée parmi un ensemble d'actions possibles (atterrir, sauter un point de passage, ...) à exécuter. Le choix de l'une ou l'autre de ces actions se base sur leur évaluation selon plusieurs critères (risques, consommation d'énergie, objectif de la mission, ...) qui déterminent leur impact sur la mission. Une particularité de ce moteur est qu'il permet d'intégrer les préférences de l'opérateur dans ses différents niveaux en utilisant des modèles d'AMCD.

Pour illustrer l'intérêt de notre moteur de décisions multi-niveaux, nous développons ensuite un simulateur de drone autonome. Ce simulateur modélise un « Thales Watchkeeper WK450 » qui doit effectuer des missions définies par une série de points de passage (waypoints). Durant la simulation d'une mission, le simulateur affiche la zone de vol en 3 dimensions, ce qui permet de visualiser l'environnement du drone. Le simulateur présente également des informations liées à la position du drone (ses coordonnées) et son état (la quantité d'énergie restante, ...). Le moteur de décisions multi-niveaux que nous proposons est utilisé par le drone pour prendre une décision en cas de besoin. Le simulateur affiche alors la liste des actions possibles ainsi que leur évaluation dans le but d'informer l'utilisateur de détails sur le déroulement de la mission.

Nous discutons ensuite de manière théorique de l'intégration de modèles d'AMCD dans les différents niveaux du moteur. Dans notre contexte applicatif, nous proposons alors d'intégrer un modèle de tri utilisant le paradigme du surclassement dans le niveau 1 (MR-Sort) et un modèle de classement du même paradigme dans le niveau 2 (SRMP). Ces modèles nécessitent de fixer leurs paramètres préférentiels en amont de la mission via un processus d'élicitation des préférences. Dans notre travail nous focalisons sur le niveau 2 et nous proposons une approche métaheuristique basée sur un algorithme génétique pour éliciter par lot les paramètres d'un modèle SRMP à partir d'un ensemble de comparaisons par paires (auxquelles l'opérateur est soumis) et qui est fourni en une seule itération à l'algorithme.

Pour terminer nous proposons un processus d'élicitation incrémentale des modèles SRMP afin de réduire l'effort cognitif de l'opérateur pendant cette phase d'apprentissage. Ce processus contient une étape de sélection d'une paire d'apprentissage à chaque itération. Nous proposons une heuristique de sélection qui permet d'atteindre un modèle SRMP satisfaisant pour l'opérateur avec un minimum de paires d'apprentissage. Nous utilisons d'abord un programme linéaire mixte en nombres entiers (MIP) pour inférer le modèle SRMP. Cependant, nous montrons que cela engendre un problème de temps d'exécution, ce qui limite l'utilisation du processus incrémental à des petits problèmes de décisions. Pour résoudre ce problème de temps de calcul, nous remplaçons le MIP par l'algorithme génétique développé plus tôt dans notre travail pour inférer un modèle SRMP satisfaisant à chaque itération du processus.

Limites et discussion

À la fin de cette thèse nous avons réussi à répondre à une grande partie des questions de recherche qui permettent d'atteindre les objectifs fixés pour la thèse. Le premier objectif est de proposer un moteur de décision à embarquer dans le drone qui garantit un niveau élevé de la confiance du pilote dans les décisions prises par le drone durant la mission. Le deuxième objectif concerne la réduction de l'effort cognitif de l'opérateur pendant la phase d'élicitation de ses préférences, qui sont embarquées dans le moteur.

L'avantage de notre moteur de décisions multi-niveaux est l'explicabilité accrue de son fonctionnement qui est garantie par la séparation des deux niveaux de décisions, tout en gardant un lien logique entre les deux. Un autre avantage est la traçabilité des décisions prises par le drone puisqu'elles sont décomposées en un ensemble de sous-décisions sur les différents niveaux de décisions du moteur. Ces éléments contribuent à augmenter la confiance de l'opérateur au sol dans les décisions du drone.

Dans ce travail nous nous sommes surtout focalisés sur les aspects liés à la prise de décision dans ce moteur. Un autre composant essentiel de ce moteur est le « contexte » du drone, qui intervient notamment dans l'évaluation des différentes alternatives des deux niveaux (les événements pour le niveau 1 et les actions possibles pour le niveau 2), ou dans leur filtrage. Dans notre travail, ce contexte est défini uniquement par la zone de vol, alors qu'il pourrait prendre en considération d'autres éléments dans l'évaluation des alternatives, comme les conditions météorologiques, ou le type de missions. Par ailleurs, dans notre travail, nous avons choisi de ne pas nous occuper des possibles défaillances des capteurs du drone. Cependant, pour rendre le moteur plus opérationnel, cet aspect est important, puisque les données de ces capteurs défaillants interviennent dans la modélisation du contexte, et par conséquent dans l'évaluation des alternatives de haut niveau.

Dans le moteur de décisions, un modèle MR-Sort est utilisé pour le niveau 1 et un modèle SRMP pour le niveau 2. Ces modèles permettent assez facilement de générer des règles qui synthétisent les raisons pour lesquelles une décision est prise. Cela répond par ailleurs à notre souci d'interprétabilité des décisions du drone et donc contribue à la confiance accrue de l'opérateur dans les décisions du drone. Ces modèles nécessitent une phase d'élicitation à partir des préférences holistiques de l'opérateur. Cependant, à l'issue de cette élicitation, les modèles compatibles avec les préférences exprimées par l'opérateur restent potentiellement nombreux. Ces différents modèles peuvent engendrer les mêmes recommandations (et donc le même comportement du drone), mais pour des raisons potentiellement différentes. Cette variabilité dans l'interprétabilité reste un souci dont nous ne nous sommes pas occupés dans ce travail.

Le simulateur de drone autonome illustre bien l'intérêt de notre modèle de décision. En effet, pendant la simulation d'une mission le drone se comporte selon les priorités et les préférences de l'opérateur élicitées en amont. Nous illustrons cela par l'exécution d'une même mission par le drone selon deux opérateurs avec des profils différents. Lors de cette mission, le drone se comporte bien différemment selon ces deux profils, et nous montrons que les décisions sont compatibles avec ces profils. Ceci contribue une nouvelle fois à la confiance de l'opérateur dans les décisions prises par le drone, étant donné que le drone a le comportement attendu par l'opérateur. Cependant, le simulateur nécessiterait une campagne de validation plus exhaustive, avec différents types de missions, et surtout une confrontation à de vrais opérateurs (qui dans nos travaux, pour des raisons de simplification, sont des modèles SRMP fixés à l'avance). Nous avons également choisi

de simplifier le niveau 1 dans cette version de simulateur, où nous considérons que l'arrivée à un waypoint est l'événement déclencheur du niveau 2.

La proposition d'un processus d'élicitation incrémentale des modèles SRMP avec un MIP combiné avec l'heuristique de sélection des paires d'apprentissage diminue le nombre nécessaire des paires pour atteindre un niveau de qualité donné du modèle des préférences. Cela sous-entend que l'opérateur est confronté à moins de questions dans la phase d'élicitation, ce qui contribue à réduire son effort cognitif, ce qui est un des objectifs de ce travail. Cependant l'étude de ce processus montre également que les temps de calcul sont trop élevés pour une élicitation interactive avec un décideur. Nous avons donc choisi d'utiliser la métaheuristique que nous avons développé à la place du MIP pour rendre le processus plus réactif. Cependant, l'utilisation de l'algorithme génétique demande un nombre plus grand de paires d'apprentissage par rapport au MIP pour atteindre un niveau de qualité donné du modèle des préférences. L'opérateur est donc soumis à plus de questions, ce qui augmente l'effort à produire. A ce niveau de nos travaux, le choix entre la méthode exacte et la méthode approchée revient donc à faire un compromis entre niveau de qualité à atteindre et temps d'exécution à chaque itération du processus incrémental.

Perspectives

Les travaux menés durant cette thèse laissent entrevoir de nombreuses perspectives de recherche intéressantes pour les différentes parties abordées dans ce manuscrit. Nous présentons ci-après celles qui nous semblent les plus importantes à explorer :

- Dans la version actuelle du moteur de décisions multi-niveaux, le contexte du drone est défini uniquement par la zone de vol. Dans des travaux futurs, nous envisageons de nous focaliser sur la modélisation de ce contexte, en enrichissant les éléments qui le définissent, comme par exemple le vent, les conditions météorologiques, le type de mission, . . .
- Afin de valider notre proposition de moteur de décisions, il devrait être soumis à une plus grande variété de missions et de contextes. Au-delà de ça, une confrontation à de vrais opérateurs et des experts serait un plus, afin de valider sa structure, ou d'identifier éventuellement d'autres niveaux de décisions complémentaires.
- Comme déjà mentionné dans notre travail, dans le simulateur, le niveau 1 du moteur de décisions est simplifié par un seul événement, l'arrivée à un waypoint, qui déclenche systématiquement le niveau 2. Une de nos perspectives est donc de développer ce niveau en prenant en considération différents événements de différents types. Des événements propres au drone, comme par exemple les pannes des capteurs, tout comme des événements extérieurs, comme l'apparition d'un obstacle, devraient faire partie de ces travaux complémentaires.
- L'environnement du drone dans le simulateur est très simple pour le moment. Il serait intéressant de modéliser un environnement de drone plus réaliste, en développant un modèle complet de vent et un modèle météorologique, soit en se basant sur une base de données d'historiques météorologiques d'une zone ou bien à partir de données temps réel.

- Dans notre travail, l'opérateur est remplacé par un modèle SRMP généré artificiellement. Nous souhaiterions confronter le simulateur à de vrais opérateurs, avec des profils différents, ce qui permettra notamment d'observer le comportement du drone selon des vrais profils d'opérateurs.
- Pour la phase d'élicitation des modèles SRMP nous confrontons l'opérateur à des paires d'actions où il doit exprimer ses préférences. Ces actions sont actuellement présentées sous la forme de vecteurs contenant les performances selon les différents critères. Il serait certainement plus intéressant et plus facile pour l'opérateur d'utiliser le simulateur pour la phase d'élicitation de ses préférences. Le simulateur dans ce cas permettra d'afficher l'environnement du drone, que l'exécution des deux actions proposées et leur évaluation sur chaque critère.
- Un algorithme génétique est proposé dans cette thèse pour éliciter par lot les modèles SRMP. Afin de mieux valider cette proposition, cet algorithme nécessiterait d'être confronté à d'autres métaheuristiques. Le but serait alors de trouver une métaheuristique qui permettrait d'atteindre un meilleur niveau de qualité que notre algorithme avec moins de paires d'apprentissage.
- Pour le processus d'élicitation des préférences incrémentale pour les modèles SRMP, nous proposons une étude des heuristiques de sélection de paires d'apprentissage à chaque itération afin de réduire l'effort cognitif de l'opérateur. Une étude plus large pourrait être intéressante où nous intégrerions d'autres heuristiques de sélection en s'appuyant sur les travaux réalisés dans le contexte des méthodes MAVT, comme par exemple une utilisation d'une règle de MinMaxRegret comme mesure de sélection.
- Pour résoudre le problème de temps de calcul pour l'élicitation incrémentale avec le MIP, nous utilisons notre algorithme génétique à la place du MIP. Actuellement nous avons utilisé le paramétrage déterminé dans notre étude d'élicitation par lot. Il pourrait cependant être intéressant de faire une étude plus approfondie sur ce paramétrage, qui produise de meilleurs résultats dans le processus d'élicitation incrémentale.
- Une autre piste qui nous paraît intéressante à étudier, et qui permettrait de résoudre partiellement le problème du temps d'exécution pour l'élicitation incrémentale est d'alterner entre une élicitation incrémentale et une élicitation par lot. Cela permettrait d'utiliser le MIP de manière plus parcimonieuse et donc de réduire le temps de calcul, tout en essayant d'offrir un bon niveau de qualité de la solution proposée.
- Toujours dans le perspective de réduire le temps de calcul pour l'élicitation incrémentale, nous pourrions également envisager de proposer un ensemble d'alternatives à l'opérateur (au lieu d'une paire) à chaque itération.
- Une toute autre voie pour diminuer le temps de calcul à chaque itération du processus incrémental serait l'accélération matérielle. Celle-ci consisterait ici à effectuer les calculs de la métaheuristique sur un circuit intégré du type FPGA ((Field-Programmable Gate Array) ou GPU (Graphics Processing Unit) dédié en vue d'exécuter cette tâche d'élicitation des préférences du modèle SRMP plus rapidement. Nous avons commencé à explorer cette voie et les résultats préliminaires sont prometteurs.

Bibliographie

- [Agence européenne pour l'environnement, 2012] AGENCE EUROPÉENNE POUR L'ENVIRONNEMENT (2012). La base de données CORINE Land Cover 2012. <https://www.data.gouv.fr/fr/datasets/corine-land-cover-occupation-des-sols-en-france/>.
- [Ajami *et al.*, 2013] AJAMI, A., BALMAT, J. F., GAUTHIER, J. P. et MAILLOT, T. (2013). Path planning and Ground Control Station simulator for UAV. In *2013 IEEE Aerospace Conference*, pages 1–13.
- [Albus *et al.*, 1988] ALBUS, J., LUMIA, R. et MCCAIN, H. (1988). Hierarchical control of intelligent machines applied to space station telerobots. *IEEE transactions on aerospace and electronic systems*, 24(5):535–541.
- [Angilella *et al.*, 2010] ANGILELLA, S., GRECO, S. et MATARAZZO, B. (2010). Non-additive robust ordinal regression : A multiple criteria decision model based on the Choquet integral. *European Journal of Operational Research*, 201(1):277 – 288.
- [Armstrong *et al.*, 1975] ARMSTRONG, J. S., DENNISTON JR, W. B. et GORDON, M. M. (1975). The use of the decomposition principle in making judgments. *Organizational behavior and human performance*, 14(2):257–263.
- [Barbier et Chanthery, 2004] BARBIER, M. et CHANTHERY, E. (2004). Autonomous mission management for unmanned aerial vehicles. *Aerospace science and technology*, 8(4):359–368.
- [Barrie, 1997] BARRIE, D. (1997). Dull, Dirty and Dangerous. *Flight International*, 151(4578):58.
- [Belahcene *et al.*, 2018] BELAHCENE, K., MOUSSEAU, V., OUERDANE, W., PIRLOT, M. et SOBRIE, O. (2018). Ranking with Multiple reference Points : Efficient Elicitation and Learning Procedures. In *Proceeding : From Multiple Criteria Decision Aid to Preference Learning (DA2PL)*.
- [Belta *et al.*, 2007] BELTA, C., BICCHI, A., EGERSTEDT, M., FRAZZOLI, E., KLAVINS, E. et PAPPAS, G. (2007). Symbolic planning and control of robot motion. *IEEE Robotics & Automation Magazine*, 14(1):61–70.
- [Belton, 1999] BELTON, V. (1999). *Multi-Criteria Problem Structuring and Analysis in a Value Theory Framework*, pages 335–366. Springer US, Boston, MA.
- [Benabbou *et al.*, 2017] BENABBOU, N., PERNY, P. et VIAPPIANI, P. (2017). Incremental elicitation of Choquet capacities for multicriteria choice, ranking and sorting problems. *Artificial Intelligence*, 246:152 – 180.
- [Betts, 1998] BETTS, J. (1998). Survey of Numerical Methods for Trajectory Optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207.
- [Beuthe et Scannella, 2001] BEUTHE, M. et SCANNELLA, G. (2001). Comparative analysis of UTA multicriteria methods. *European Journal of Operational Research*, 130(2):246–262.

- [Bisdorff *et al.*, 2015] BISDORFF, R., DIAS, L., MEYER, P., MOUSSEAU, V. et PIRLOT, M. (2015). *Evaluation and Decision Models With Multiple Criteria : Case Studies*. International Handbooks on Information Systems. Springer.
- [Bisdorff *et al.*, 2008] BISDORFF, R., MEYER, P. et ROUBENS, M. (2008). RUBIS : a bipolar-valued outranking method for the choice problem. *4OR*, 6:143–165.
- [Borrelli *et al.*, 2006] BORRELLI, F., SUBRAMANIAN, D., RAGHUNATHAN, A. et BIEGLER, L. (2006). MILP and NLP techniques for centralized trajectory planning of multiple unmanned air vehicles. In *2006 American Control Conference*, page 6. IEEE.
- [Bouyssou et Marchant, 2007a] BOUYSSOU, D. et MARCHANT, T. (2007a). An axiomatic approach to noncompensatory sorting methods in MCDM, I : The case of two categories. *European Journal of Operational Research*, 178(1):217 – 245.
- [Bouyssou et Marchant, 2007b] BOUYSSOU, D. et MARCHANT, T. (2007b). An axiomatic approach to noncompensatory sorting methods in MCDM, II : More than two categories. *European Journal of Operational Research*, 178(1):246 – 276.
- [Bouyssou *et al.*, 2006] BOUYSSOU, D., MARCHANT, T., PIRLOT, M., TSOUKIAS, A. et VINCKE, P. (2006). *Evaluation and Decision Models with Multiple Criteria : Stepping stones for the analyst*. International Series in Operations Research & Management Science. Springer US.
- [Brans, 1982] BRANS, J. (1982). *L'ingénierie de la décision : l'élaboration d'instruments d'aide à la décision*. Université Laval, Faculté des sciences de l'administration.
- [Brans et Mareschal, 1992] BRANS, J. et MARESCHAL, B. (1992). PROMETHEE V : MCDM problems with segmentation constraints. *INFOR : Information Systems and Operational Research*, 30(2):85–96.
- [Brans et Mareschal, 1994] BRANS, J. et MARESCHAL, B. (1994). The PROMCALC & GAIA decision support system for multicriteria decision aid. *Decision support systems*, 12(4-5):297–310.
- [Butler *et al.*, 1997] BUTLER, J., JIA, J. et DYER, J. (1997). Simulation techniques for the sensitivity analysis of multi-criteria decision models. *European Journal of Operational Research*, 103(3):531 – 546.
- [Campbell *et al.*, 2013] CAMPBELL, S., BRAGG, M. et NEOGI, N. (2013). Fuel-Optimal Trajectory Generation for Persistent Contrail Mitigation. *Journal of Guidance, Control, and Dynamics*, 36(6):1741–1750.
- [Chandler et Pachter, 1998] CHANDLER, P. et PACTER, M. (1998). Research issues in autonomous control of tactical UAVs. In *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No. 98CH36207)*, volume 1, pages 394–398. IEEE.
- [Choquet, 1954] CHOQUET, G. (1954). Theory of capacities. *Annales de l'Institut Fourier*, 5:131–295.
- [Ciomek *et al.*, 2017] CIOMEK, K., KADZIŃSKI, M. et TERVONEN, T. (2017). Heuristics for selecting pair-wise elicitation questions in multiple criteria choice problems. *European Journal of Operational Research*, 262(2):693 – 707.
- [Clough, 2002] CLOUGH, B. (2002). Metrics, Schmetrics! How Do You Track a UAV's Autonomy? In *1st UAV Conference*, page 3499.
- [Dalamagkidis, 2015] DALAMAGKIDIS, K. (2015). *Classification of UAVs*, pages 83–91. Springer Netherlands, Dordrecht.
- [Dias et Clímaco, 2000] DIAS, L. et CLÍMACO, J. (2000). ELECTRE TRI for Groups with Imprecise Information on Parameter Values. *Group Decision and Negotiation*, 9(5):355–377.

- [Dias et Mousseau, 2018] DIAS, L. C. et MOUSSEAU, V. (2018). *Eliciting Multi-Criteria Preferences : ELECTRE Models*, pages 349–375. Springer International Publishing, Cham.
- [Dodge, 2008] DODGE, Y. (2008). *Kolmogorov–Smirnov Test*, pages 283–287. Springer New York, New York, NY.
- [Durbach, 2009] DURBACH, I. (2009). The use of the SMAA acceptability index in descriptive decision analysis. *European Journal of Operational Research*, 196(3):1229 – 1237.
- [Ergezer et Leblebicioglu, 2013] ERGEZER, H. et LEBLEBICIOGLU, K. (2013). Path Planning for UAVs for Maximum Information Collection. *IEEE Transactions on Aerospace and Electronic Systems*, 49(1):502–520.
- [FAA, 1991] FAA (1991). Federal Aviation Administration, Advisory Circular, Aeronautical Decision making. *Aeronautical Decision Making*.
- [Fadok, 1995] FADOK, D. (1995). John Boyd and John Warden : Air Power’s Quest for Strategic Paralysis. Rapport technique, Air Univ, Maxwell AFB, AL, School of Advanced Airpower Studies.
- [Fan et al., 2015] FAN, J., ZHANG, J., MEI, K., PENG, J. et GAO, L. (2015). Cost-sensitive Learning of Hierarchical Tree Classifiers for Large-scale Image Classification and Novel Category Detection. *Pattern Recogn.*, 48(5):1673–1687.
- [Figueira et al., 2004] FIGUEIRA, J., DE SMET, Y. et BRANS, J. (2004). MCDA methods for sorting and clustering problems : Promethee TRI and Promethee CLUSTER. *Université Libre de Bruxelles. Service de Mathématiques de la Gestion, Working Paper*, 2.
- [Figueira et al., 2005] FIGUEIRA, J., GRECO, S. et EHRGOTT, M. (2005). *Multiple Criteria Decision Analysis : State of the Art Surveys*. International Series in Operations Research & Management Science. Springer.
- [Figueira et al., 2010] FIGUEIRA, J., GRECO, S., ROY, B. et SŁOWIŃSKI, R. (2010). *ELECTRE Methods : Main Features and Recent Developments*, pages 51–89. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Fishburn, 1964] FISHBURN, P. (1964). Decision and value theory.
- [Fishburn, 1965] FISHBURN, P. (1965). Independence in utility theory with whole product sets. *Operations Research*, 13(1):28–45.
- [Fishburn, 1970] FISHBURN, P. (1970). Utility theory for decision making.
- [Franco et Buttazzo, 2015] FRANCO, C. D. et BUTTAZZO, G. (2015). Energy-Aware Coverage Path Planning of UAVs. In *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*, pages 111–117.
- [Frazzoli et al., 2002] FRAZZOLI, E., DAHLEH, M. A. et FERON, E. (2002). Real-Time Motion Planning for Agile Autonomous Vehicles. *Journal of Guidance, Control, and Dynamics*, 25(1): 116–129.
- [Gholipour et al., 2013] GHOLIPOUR, A., HOSSEINI, M. J. et BEIGY, H. (2013). An Adaptive Regression Tree for Non-stationary Data Streams. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13*, pages 815–817, New York, NY, USA. ACM.
- [Gigerenzer et Todd, 1999] GIGERENZER, G. et TODD, P. M. (1999). *Simple heuristics that make us smart*. Oxford University Press, USA.
- [Goerzen et al., 2009] GOERZEN, C., KONG, Z. et METTLER, B. (2009). A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance. *Journal of Intelligent and Robotic Systems*, 57(1):65.

- [Grabisch et Labreuche, 2010] GRABISCH, M. et LABREUCHE, C. (2010). A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid. *Annals of Operations Research*, 175(1):247–286.
- [Guitouni et Martel, 1998] GUITOUNI, A. et MARTEL, J. (1998). Tentative guidelines to help choosing an appropriate MCDA method. *European Journal of Operational Research*, 109(2): 501–521.
- [Holland, 1975] HOLLAND, J. H. (1975). *Adaptation in natural and artificial systems*. The University of Michigan Press.
- [Holloway et White III, 2003] HOLLOWAY, H. et WHITE III, C. C. (2003). Question selection for multi-attribute decision-aiding. *European Journal of Operational Research*, 148(3):525 – 533.
- [Huang, 2004] HUANG, H. (2004). Autonomy levels for unmanned systems (ALFUS) framework volume I : Terminology version 2.0.
- [Hugonnard et Roy, 1982] HUGONNARD, J. et ROY, B. (1982). Le plan d’extension du métro en banlieue parisienne, un cas type d’application de l’analyse multicritère . *Cahiers Scientifiques de la Revue Transports*, (6).
- [IBM ILOG , 2018] IBM ILOG (2018). CPLEX Optimization Studio, Version 12.8.
- [INSEE, 2010] INSEE (2010). Estimations carroyées de population. <https://www.insee.fr/fr/statistiques/2520034>.
- [Ishizaka et Nemery, 2013] ISHIZAKA, A. et NEMERY, P. (2013). *Multi-criteria decision analysis : methods and software*. John Wiley & Sons.
- [Israelsen et al., 2014] ISRAELSEN, J., BEALL, M., BAREISS, D., STUART, D., KEENEY, E. et van den BERG, J. (2014). Automatic collision avoidance for manually tele-operated unmanned aerial vehicles. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6638–6643. IEEE.
- [Jacquet-Lagreze et Siskos, 1982] JACQUET-LAGREZE, E. et SISKOS, J. (1982). Assessing a set of additive utility functions for multicriteria decision-making, the UTA method. *European Journal of Operational Research*, 10(2):151 – 164.
- [Jain et Tsiotras, 2008] JAIN, S. et TSIOTRAS, P. (2008). Trajectory Optimization Using Multi-resolution Techniques. *Journal of Guidance, Control, and Dynamics*, 31(5):1424–1436.
- [Jensen, 2007] JENSEN, J. (2007). Remote sensing of the environment : an earth resource perspective : Pearson Prentice Hall. *Upper Saddle River, NJ*.
- [Ju et Son, 2018] JU, C. et SON, H. (2018). Multiple UAV systems for agricultural applications : control, implementation, and evaluation. *Electronics*, 7(9):162.
- [Keeney et Raiffa, 1976] KEENEY, R. L. et RAIFFA, H. (1976). *Decision with Multiple Objectives*. Wiley, New York.
- [Keeney et Raiffa, 1993] KEENEY, R. L. et RAIFFA, H. (1993). *Decisions with Multiple Objectives : Preferences and Value Trade-Offs*. Wiley series in probability and mathematical statistics. Applied probability and statistics. Cambridge University Press.
- [Kendall, 1938] KENDALL, M. G. (1938). A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.
- [Ketema et Zhao, 2010] KETEMA, Y. et ZHAO, Y. (2010). Controllability and reachability for micro-aerial-vehicle trajectory planning in winds. *Journal of Guidance, Control, and Dynamics*, 33(3):1020–1023.

- [Khannoussi *et al.*, 2017] KHANNOUSSI, A., MEYER, P., DEZAN, C. et DIGUET, J.-P. (2017). Integrating human preferences in automated decisions of unmanned aerial vehicles. *In 14th Decision Deck Workshop*.
- [Khannoussi *et al.*, 2018a] KHANNOUSSI, A., MEYER, P., DEZAN, C., DIGUET, J.-P. et OLTEANU, A.-L. (2018a). Traceable decisions for autonomous unmanned aerial vehicles. *In ROADEF 2018*.
- [Khannoussi *et al.*, 2018b] KHANNOUSSI, A., OLTEANU, A.-L., DEZAN, C., DIGUET, J.-P., LABREUCHE, C., PETIT-FRÈRE, J. et MEYER, P. (2018b). Incremental learning of simple ranking method using reference profiles models. *In DA2PL'2018 : from Multiple Criteria Decision Aid to Preference Learning*.
- [Khannoussi *et al.*, 2018c] KHANNOUSSI, A., OLTEANU, A.-L., DEZAN, C., DIGUET, J.-P. et MEYER, P. (2018c). Incremental preference elicitation for srmp models : Application for autonomous drones. *In 88th Meeting of the EURO Working Group Multiple Criteria Decision Aiding*.
- [Khannoussi *et al.*, 2019] KHANNOUSSI, A., OLTEANU, A.-L., LABREUCHE, C., NARAYAN, P., DEZAN, C., DIGUET, J.-P., PETIT-FRÈRE, J. et MEYER, P. (2019). Integrating operators' preferences into decisions of unmanned aerial vehicles : Multi-layer decision engine and incremental preference elicitation. *In International Conference on Algorithmic Decision Theory*, pages 49–64. Springer.
- [Kojadinovic, 2007] KOJADINOVIC, I. (2007). Minimum variance capacity identification. *European Journal of Operational Research*, 177(1):498–514.
- [Kolitz, 1993] KOLITZ, S. (1993). Computing techniques in mission planning. *AGARD, New Advances in Mission Planning and Rehearsal Systems 20 p(SEE N 94-25008 06-66)*.
- [Lahdelma *et al.*, 1998] LAHDELMA, R., HOKKANEN, J. et SALMINEN, P. (1998). SMAA - stochastic multiobjective acceptability analysis. *European Journal of Operational Research*, 106(1):137 – 143.
- [LaValle, 1998] LAVALLE, S. (1998). Rapidly-exploring random trees : A new tool for path planning.
- [Leroy *et al.*, 2011] LEROY, A., MOUSSEAU, V. et PIRLOT, M. (2011). Learning the Parameters of a Multiple Criteria Sorting Method. *In Proceedings of the Second International Conference on Algorithmic Decision Theory, ADT'11*, pages 219–233, Berlin, Heidelberg. Springer-Verlag.
- [Liu *et al.*, 2014] LIU, J., OUERDANE, W. et MOUSSEAU, V. (2014). Metaheuristic approach for preference learning in multicriteria ranking based on reference points. *In Proceeding : of the 2nd workshop From Multiple Criteria Decision aid to Preference Learning (DA2PL)*.
- [Marshall, 2004] MARSHALL, D. (2004). Dull, Dirty, and Dangerous : The FAA's Regulatory Authority over Unmanned Aircraft Operations. *Issues Aviation L. & Pol'y*, page 10085.
- [Marshall *et al.*, 2016] MARSHALL, D., BARNHART, R., HOTTMAN, S., SHAPPEE, E. et MOST, M. (2016). *Introduction to unmanned aircraft systems*. Crc Press.
- [Massaglia et Ostanello, 1991] MASSAGLIA, R. et OSTANELLO, A. (1991). N-tomic : a support system for multicriteria segmentation problems. *Multiple Criteria Decision Support*, 356:167–174.
- [MATLAB, 2018] MATLAB (2018). *version 9.4.0.813654 (R2018a)*. The Mathworks, Inc., Natick, Massachusetts.
- [Maystre *et al.*, 1994] MAYSTRE, L. Y., PICTET, J. et SIMOS, J. (1994). *Méthodes multicritères ELECTRE : description, conseils pratiques et cas d'application à la gestion environnementale*, volume 8. PPUR Presses Polytechniques.

- [Maza *et al.*, 2009] MAZA, I., KONDAK, K., BERNARD, M. et OLLERO, A. (2009). Multi-UAV cooperation and control for load transportation and deployment. *In Selected papers from the 2nd International Symposium on UAVs, Reno, Nevada, USA June 8–10, 2009*, pages 417–449. Springer.
- [McGee et Hedrick, 2007] MCGEE, T. et HEDRICK, J. (2007). Optimal Path Planning with a Kinematic Airplane Model. *Journal of Guidance, Control, and Dynamics*, 30(2):629–633.
- [Meyer et Roubens, 2006] MEYER, P. et ROUBENS, M. (2006). On the use of the Choquet integral with fuzzy numbers in multiple criteria decision support. *Fuzzy Sets and Systems*, 157(7):927–938.
- [Mikaelian *et al.*, 2012] MIKAEKIAN, T., RHODES, D. H., NIGHTINGALE, D. J. et HASTINGS, D. E. (2012). A Logical Approach to Real Options Identification With Application to UAV Systems. *IEEE Transactions on Systems, Man, and Cybernetics - Part A : Systems and Humans*, 42(1):32–47.
- [Mori, 1989] MORI, T. (1989). An analysis of evaluation model using fuzzy measure and the Choquet integral. *In Proc. 5th Fuzzy systems symposium*, pages 207–212.
- [Moscarola et Roy, 1977] MOSCAROLA, J. et ROY, B. (1977). Procédure automatique d'examen de dossiers fondée sur une segmentation trichotomique en présence de critères multiples. *RAIRO-Operations Research*, 11(2):145–173.
- [Mousseau, 2005] MOUSSEAU, V. (2005). A general framework for constructive learning preference elicitation in multiple criteria decision aid.
- [Mousseau *et al.*, 2003] MOUSSEAU, V., FIGUEIRA, J., DIAS, L., da SILVA, C. G. et CLÍMACO, J. (2003). Resolving inconsistencies among constraints on the parameters of an MCDA model. *European Journal of Operational Research*, 147(1):72–93.
- [Mousseau *et al.*, 2001] MOUSSEAU, V., FIGUEIRA, J. et NAUX, J. (2001). Using assignment examples to infer weights for ELECTRE TRI method : Some experimental results. *European Journal of Operational Research*, 130.
- [Mousseau et Slowinski, 1998] MOUSSEAU, V. et SLOWINSKI, R. (1998). Inferring an ELECTRE TRI Model from Assignment Examples. *Journal of Global Optimization*, 12(2):157–174.
- [Mousseau *et al.*, 2000] MOUSSEAU, V., SLOWINSKI, R. et ZIELNIEWICZ, P. (2000). A user-oriented implementation of the ELECTRE-TRI method integrating preference elicitation support. *Computers & OR*, 27:757–777.
- [Narayan *et al.*, 2013] NARAYAN, P., MEYER, P. et CAMPBELL, D. (2013). Embedding Human Expert Cognition Into Autonomous UAS Trajectory Planning. *IEEE Transactions on Cybernetics*, 43(2):530–543.
- [Narayan *et al.*, 2008] NARAYAN, P., WU, P. et CAMPBELL, D. (2008). Unmanning UAVs—addressing challenges in on-board planning and decision making. *the 1st International Conference on Humans Operating Unmanned Systems, Telecom Bretagne, Brest, France, 2008*.
- [Nikolos *et al.*, 2003] NIKOLOS, I. K., VALAVANIS, K. P., TSOURVELOUDIS, N. C. et KOSTARAS, A. N. (2003). Evolutionary algorithm based offline/online path planner for UAV navigation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 33(6):898–912.
- [Obermeyer *et al.*, 2012] OBERMEYER, K., OBERLIN, P. et DARBHA, S. (2012). Sampling-Based Path Planning for a Visual Reconnaissance Unmanned Air Vehicle. *Journal of Guidance, Control, and Dynamics*, 35(2):619–631.
- [Olteanu, 2018] OLTEANU, A. (2018). Strategies for the incremental inference of majority-rule sorting models. in *Proceeding from multiple criteria Decision aid to Preference Learning (DA2PL)*.

- [Olteanu et Meyer, 2014] OLTEANU, A. et MEYER, P. (2014). Inferring the parameters of a majority rule sorting model with vetoes on large datasets. *In DA2PL 2014 : From Multicriteria Decision Aid to Preference Learning*, pages 87–94.
- [Olteanu et al., 2018] OLTEANU, A.-L., BELAHCENE, K., MOUSSEAU, V., OUERDANE, W., ROLLAND, A. et ZHENG, J. (2018). Preference Elicitation for a Ranking Method based on Multiple Reference Profiles. working paper or preprint.
- [Pham et al., 2008] PHAM, T., CIRINCIONE, G., VERMA, D. et PEARSON, G. (2008). Intelligence, surveillance, and reconnaissance fusion for coalition operations. *In 2008 11th International Conference on Information Fusion*, pages 1–8. IEEE.
- [Rocha et Dias, 2008] ROCHA, C. et DIAS, L. (2008). An algorithm for ordinal sorting based on ELECTRE with categories defined by examples. *Journal of Global Optimization*, 42(2):255–277.
- [Roemer et Tang, 2015] ROEMER, M. J. et TANG, L. (2015). *Integrated Vehicle Health and Fault Contingency Management for UAVs*, pages 999–1025. Springer Netherlands, Dordrecht.
- [Rogers et Bruen, 1998] ROGERS, M. et BRUEN, M. (1998). Choosing realistic values of indifference, preference and veto thresholds for use with environmental criteria within ELECTRE. *European Journal of Operational Research*, 107(3):542 – 551.
- [Roldán et al., 2015] ROLDÁN, J. J., DEL CERRO, J. et BARRIENTOS, A. (2015). A proposal of methodology for multi-UAV mission modeling. *In 2015 23rd Mediterranean Conference on Control and Automation (MED)*, pages 1–7.
- [Rolland, 2013a] ROLLAND, A. (2013a). Reference-based preferences aggregation procedures in multi-criteria decision making. *European Journal of Operational Research*, 225(3):479–486.
- [Rolland, 2013b] ROLLAND, A. (2013b). Reference-based preferences aggregation procedures in multi-criteria decision making. *European Journal of Operational Research*, 225(3):479–486.
- [Roubens et Vincke, 1985] ROUBENS, M. et VINCKE, P. (1985). *Preference Modeling*. LNEMS 250, Springer Verlag, Berlin.
- [Roy, 1968] ROY, B. (1968). Classement et choix en présence de points de vue multiples. *Revue française d’informatique et de recherche opérationnelle*, 2(8):57–75.
- [Roy, 1971] ROY, B. (1971). La méthode electre ii, metra, direction scientifique. *Note de Travail No, 142*.
- [Roy, 1978] ROY, B. (1978). ELECTRE III : Un algorithme de classement fondé sur une représentation floue des préférences en présence de critères multiples. *Cahiers du CERO*, 20(1):3–24.
- [Roy, 1985] ROY, B. (1985). *Méthodologie Multicritère d’Aide la Décision*. Economica.
- [Roy, 1990] ROY, B. (1990). The outranking approach and the foundations of electre methods. *In Bana e COSTA, C. A., éditeur : Readings in Multiple Criteria Decision Aid*, pages 155–183, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Roy, 1991] ROY, B. (1991). The Outranking Approach and the Foundations of Electre Methods. *Theory and Decision*, 31(1):49–73.
- [Roy, 1996] ROY, B. (1996). *Multicriteria Methodology for Decision Aiding*. Kluwer Academic, Dordrecht.
- [Roy et Bouyssou, 1993] ROY, B. et BOUYSSOU, D. (1993). *Aide Multicritère à la Décision : Méthodes et Cas*. Paris : Economica.
- [Roy et Skalka, 1984] ROY, B. et SKALKA, J. M. (1984). ELECTRE IS : aspects méthodologiques et guide d’utilisation. *Technical Report., Document du LAMSADE, Université Paris-Dauphine, Paris*.

- [Roy et Vincke, 1987] ROY, B. et VINCKE, P. (1987). Pseudo-orders : Definition, properties and numerical representation. *Mathematical Social Sciences*, 14(3):263 – 274.
- [Ruz et al., 2006] RUZ, J. J., AREVALO, O., de la CRUZ, J. M. et PAJARES, G. (2006). Using MILP for UAVs Trajectory Optimization under Radar Detection Risk. pages 957–960.
- [Saha et al., 2011] SAHA, B., KOSHIMOTO, E., QUACH, C. C., HOGGE, E. F., STROM, T. H., HILL, B. L., VAZQUEZ, S. L. et GOEBEL, K. (2011). Battery health management system for electric UAVs. *In 2011 Aerospace Conference*, pages 1–9.
- [Salperwyck et Lemaire, 2013] SALPERWYCK, C. et LEMAIRE, V. (2013). Incremental decision tree based on order statistics. *In The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.
- [Schmeidler, 1986] SCHMEIDLER, D. (1986). Integral Representation Without Additivity. *Proceedings of The American Mathematical Society - PROC AMER MATH SOC*, 97:255–255.
- [Schrijver, 1998] SCHRIJVER, A. (1998). *Theory of linear and integer programming*. John Wiley & Sons.
- [SD, 2005] SD (2005). Secretary of Defense, Unmanned Aerial Vehicles roadmap.
- [Shom et IMT Atlantique, 2017] SHOM et IMT ATLANTIQUE (2017). DESEASON. <http://recherche.imt-atlantique.fr/deseasion/>.
- [Siskos et al., 2005] SISKOS, Y., GRIGOROUDIS, E. et MATSATSINIS, N. (2005). UTA Methods. *In FIGUEIRA, J., GRECO, S. et EHRGOTT, M., éditeurs : Multiple Criteria Decision Analysis : State of the Art Surveys*, pages 297–344. Springer Verlag, Boston, Dordrecht, London.
- [Siskos et Yanacopoulos, 1985] SISKOS, Y. et YANACOPOULOS, D. (1985). UTA STAR - an ordinal regression method for building additive value functions. *Investigacao Operational*, 5:39–53.
- [Sobrie et al., 2013] SOBRIE, O., MOUSSEAU, V. et PIRLOT, M. (2013). Learning a majority rule model from large sets of assignment examples. *In International Conference on Algorithmic Decision Theory*, pages 336–350. Springer.
- [Sobrie et al., 2017] SOBRIE, O., MOUSSEAU, V. et PIRLOT, M. (2017). A population-based algorithm for learning a majority rule sorting model with coalitional veto. *In International Conference on Evolutionary Multi-Criterion Optimization*, pages 575–589. Springer.
- [Spitzer, 2001] SPITZER, C. (2001). *Digital avionics systems - Principles and practices 2nd Edition*. The Blackburn Press.
- [Sugeno, 1974] SUGENO, M. (1974). *Theory of fuzzy integrals and its applications*. Thèse de doctorat, Tokyo Institute of Technology.
- [Syed et al., 2017] SYED, S. S., AAYUSH, A. et KAUSHIK, R. (2017). Incremental Learning in Deep Convolutional Neural Networks Using Partial Network Sharing. *CoRR*, abs/1712.02719.
- [Thales, 2010] THALES (2010). Watchkeeper. <https://www.thalesgroup.com/en/global/activities/defence/unmanned-aerial-vehicles-systems/tactical-uav>.
- [Tsoukiàs, 2006] TSOUKIÀS, A. (2006). De la théorie de la décision à l'aide à la décision. *D. Bouyssou, D. Dubois, M. Pirlot, & H. Prade, Concepts et méthodes pour l'aide à la décision*, 1.
- [Tsoukiàs, 2007] TSOUKIÀS, A. (2007). On the concept of decision aiding process : an operational perspective. *Annals of Operations Research*, 154(1):3–27.
- [Waharte et Trigoni, 2010] WAHARTE, S. et TRIGONI, N. (2010). Supporting search and rescue operations with UAVs. *In 2010 International Conference on Emerging Security Technologies*, pages 142–147. IEEE.

- [Wan et al., 2014] WAN, J., ZHONG, L. et ZHANG, F. (2014). Cooperative Localization of Multi-UAVs via Dynamic Nonparametric Belief Propagation under GPS Signal Loss Condition. *International Journal of Distributed Sensor Networks*, 10(2):562380.
- [Wang et Li, 2008] WANG, J. et LI, C. (2008). Acquisition of UAV images and the application in 3D city modeling. In *International Symposium on Photoelectronic Detection and Imaging 2007 : Image Processing*, volume 6623, page 66230Z. International Society for Optics and Photonics.
- [Wang et Ruhe, 2007] WANG, Y. et RUHE, G. (2007). The cognitive process of decision making. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 1(2):73–85.
- [Weber, 1987] WEBER, M. (1987). Decision making with incomplete information. *European Journal of Operational Research*, 28(1):44 – 57.
- [Wen et Lu, 2007] WEN, Y. et LU, B. (2007). Incremental Learning of Support Vector Machines by Classifier Combining. In ZHOU, Z., LI, H. et YANG, Q., éditeurs : *Advances in Knowledge Discovery and Data Mining*, pages 904–911, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Winter et al., 1996] WINTER, G., PERIAUX, J., GALAN, M. et CUESTA, P. (1996). *Genetic algorithms in engineering and computer science*. John Wiley & Sons, Inc.
- [Yager, 1988] YAGER, R. (1988). On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on systems, Man, and Cybernetics*, 18(1):183–190.
- [Yokoyama et Ochi, 2009] YOKOYAMA, N. et OCHI, Y. (2009). Path Planning Algorithms for Skid-to-Turn Unmanned Aerial Vehicles. *Journal of Guidance, Control, and Dynamics*, 32(5): 1531–1543.
- [Yu et Sycara, 2006] YU, B. et SYCARA, K. (2006). Learning the Quality of Sensor Data in Distributed Decision Fusion. In *2006 9th International Conference on Information Fusion*, pages 1–8.
- [Yu et al., 2010] YU, H., BEARD, R. et BYRNE, J. (2010). Vision-based navigation frame mapping and planning for collision avoidance for miniature air vehicles. *Control Engineering Practice*, 18(7):824 – 836. Special Issue on Aerial Robotics.
- [Yu, 1992] YU, W. (1992). ELECTRE TRI : aspects méthodologiques et manuel d’utilisation. *Document du LAMSADE, Université de Paris-Dauphine*.
- [Yu et Zhang, 2015] YU, X. et ZHANG, Y. (2015). Sense and avoid technologies with applications to unmanned aircraft systems : Review and prospects. *Progress in Aerospace Sciences*, 74:152 – 166.
- [Zermani et al., 2016] ZERMANI, S., DEZAN, C., HIRECHE, C., EULER, R. et DIGUET, J. (2016). Embedded and probabilistic health management for the GPS of autonomous vehicles. In *2016 5th Mediterranean Conference on Embedded Computing (MECO)*, pages 401–404.
- [Zhong et Cassandras, 2015] ZHONG, M. et CASSANDRAS, C. (2015). *Distributed Optimization of Autonomous UAVs with Event-Driven Communication*, pages 1749–1773. Springer Netherlands, Dordrecht.
- [Zhu, 1991] ZHU, Q. (1991). Hidden Markov model for dynamic obstacle avoidance of mobile robot navigation. *IEEE Transactions on Robotics and Automation*, 7(3):390–397.
- [Zopounidis et Doumpos, 1999] ZOPOUNIDIS, C. et DOUMPOS, M. (1999). A Multicriteria Decision Aid Methodology for Sorting Decision Problems : The Case of Financial Distress. *Computational Economics*, 14(3):197–218.

Annexes

Annexe A

Résultats détaillés de l'algorithme génétique

A.1 Résultats des test de la configuration 1

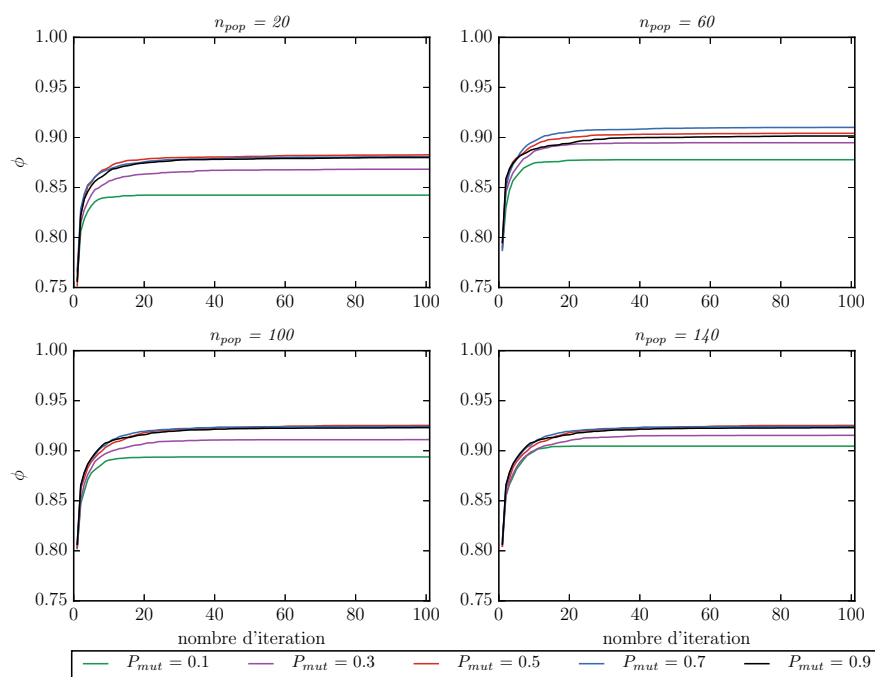
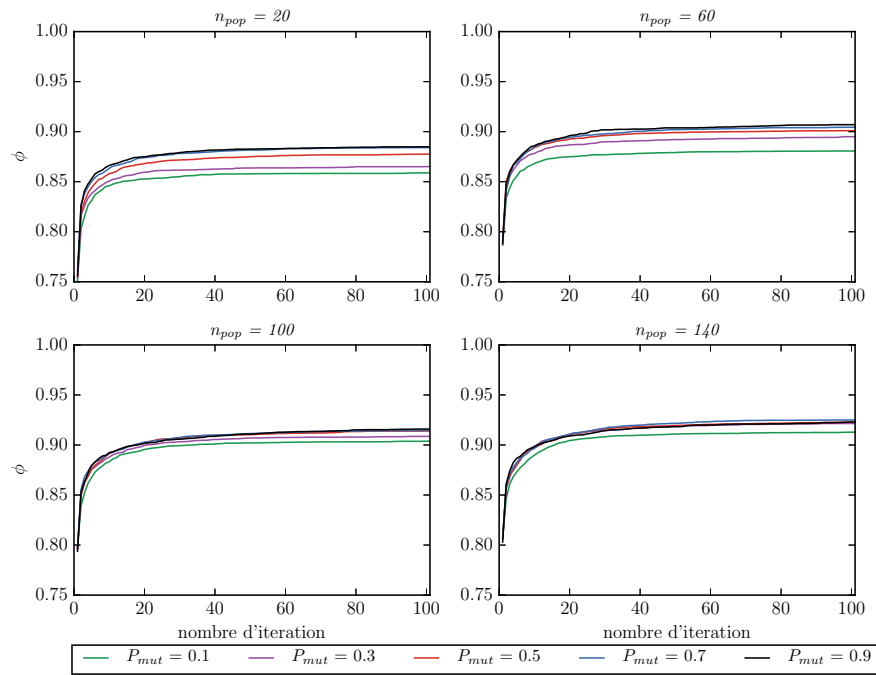
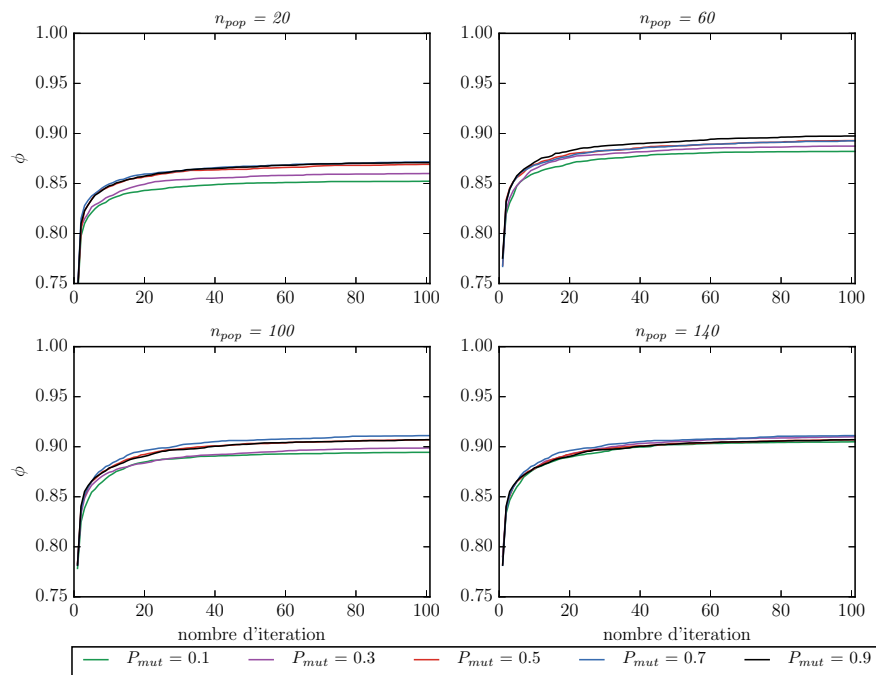
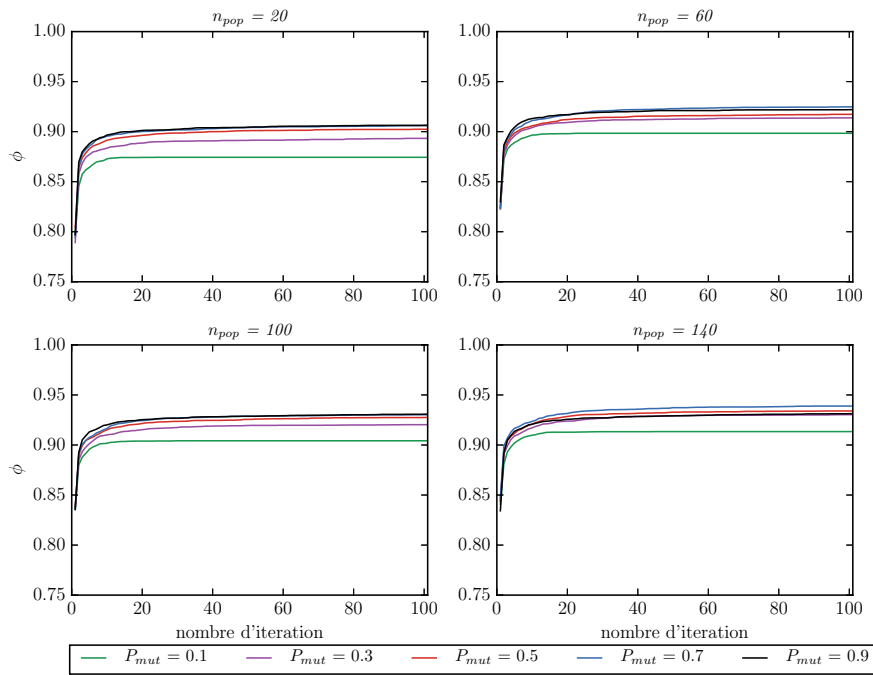
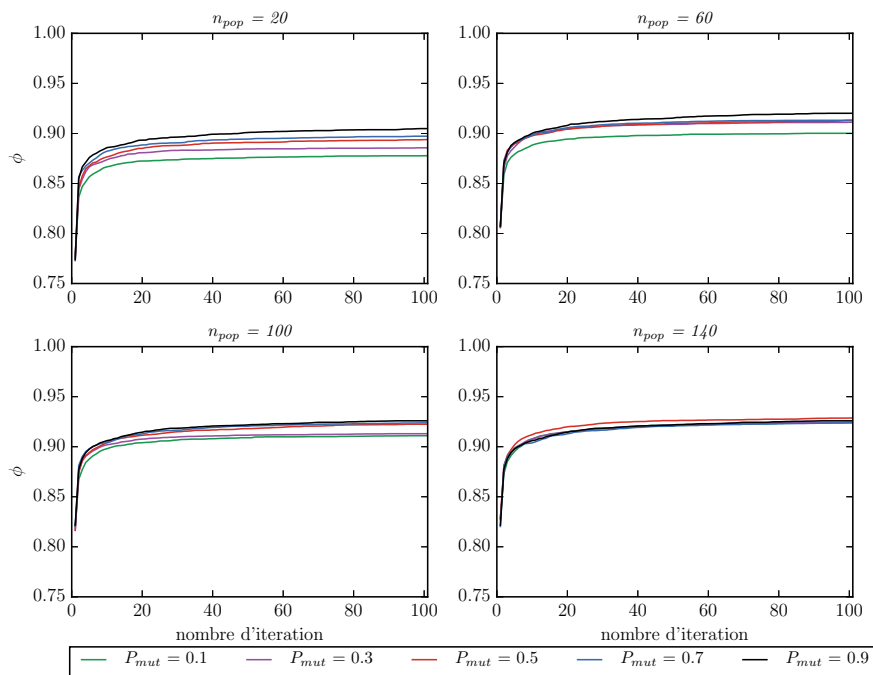
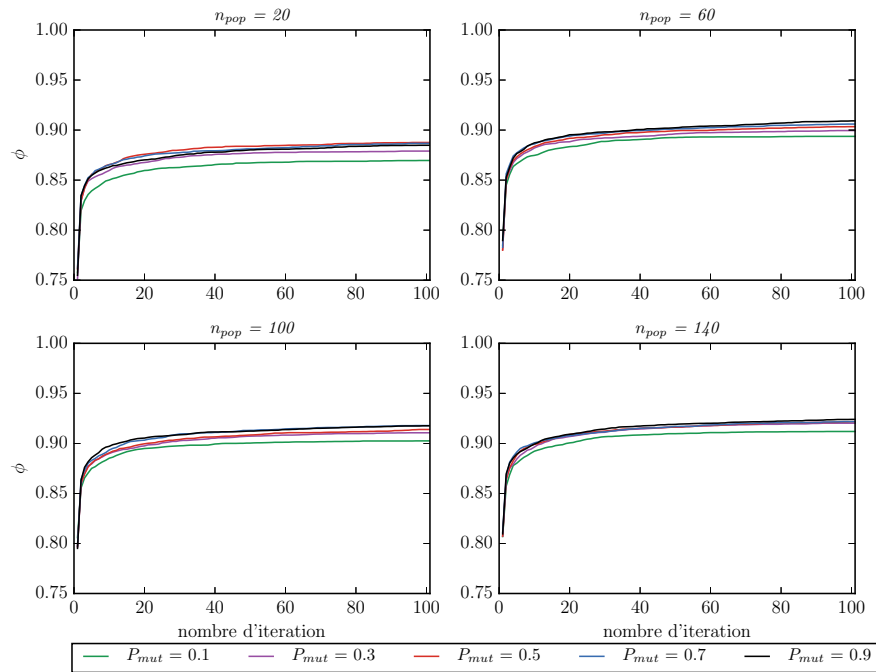


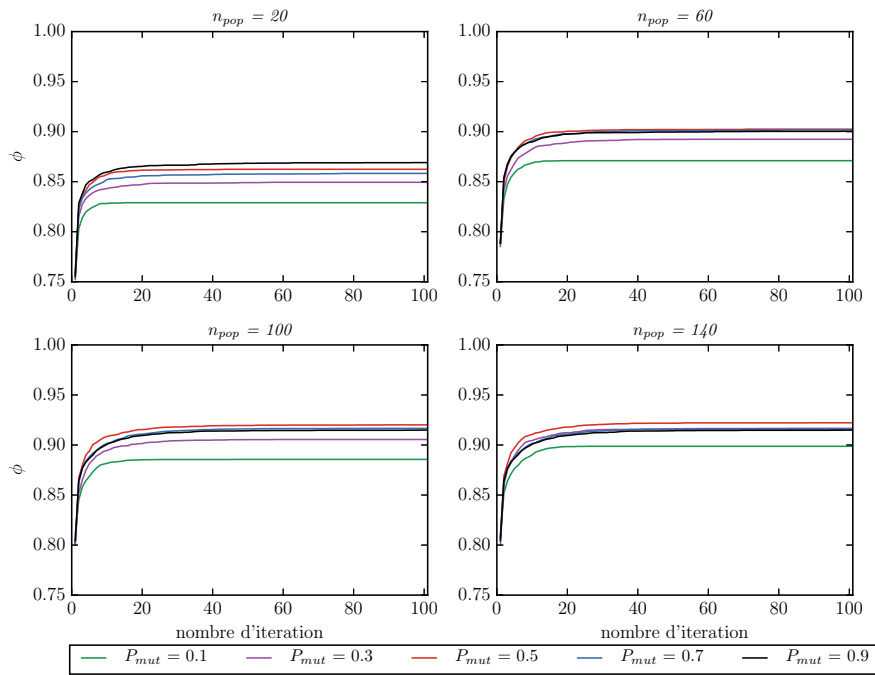
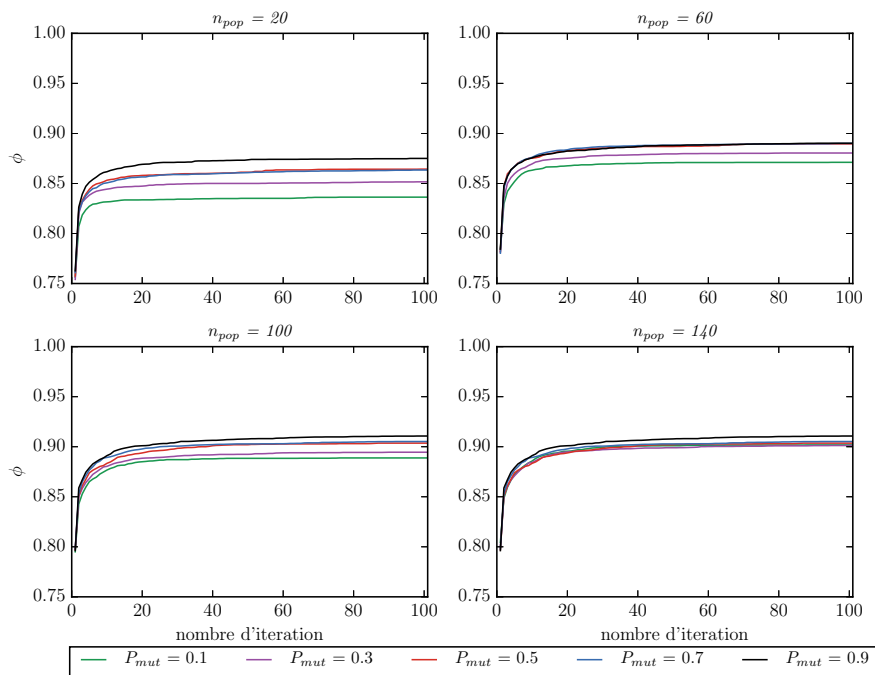
Figure A.1 – Évaluation (ϕ) moyenne pour les problèmes $2P3C$

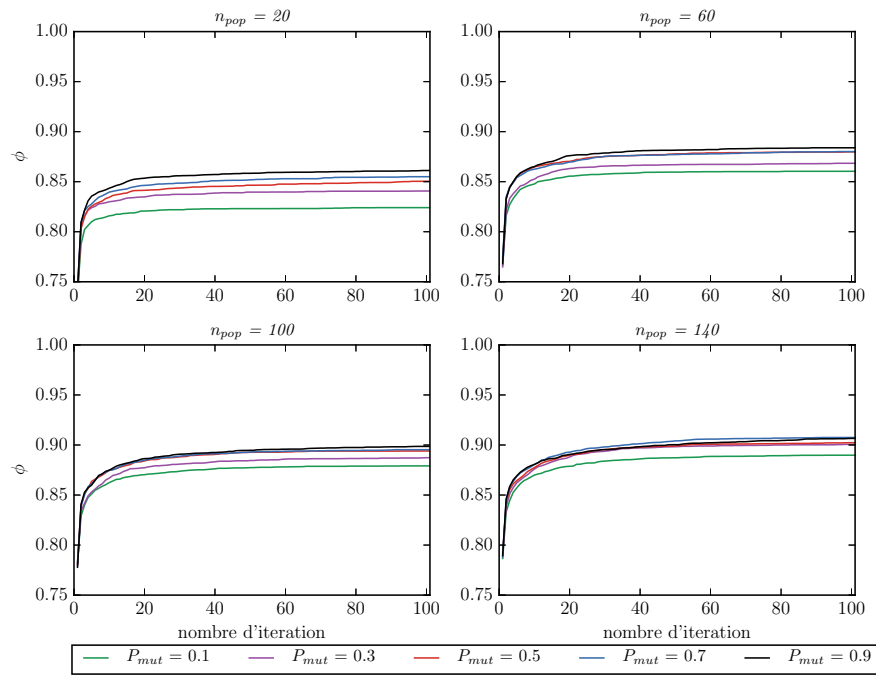
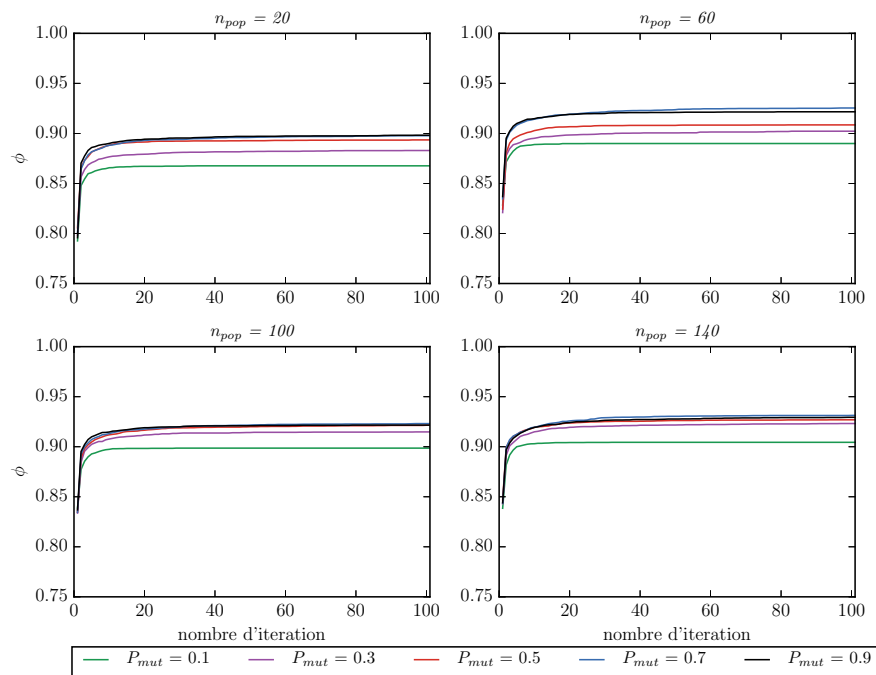
Figure A.2 – Évaluation (ϕ) moyenne pour les problèmes $2P5C$ Figure A.3 – Évaluation (ϕ) moyenne pour les problèmes $2P7C$

Figure A.4 – Évaluation (ϕ) moyenne pour les problèmes $3P3C$ Figure A.5 – Évaluation (ϕ) moyenne pour les problèmes $3P5C$

Figure A.6 – Évaluation (ϕ) moyenne pour les problèmes $3P7C$

A.2 Résultats des test de la configuration 2

Figure A.7 – Évaluation (ϕ) moyenne pour les problèmes $2P3C$ Figure A.8 – Évaluation (ϕ) moyenne pour les problèmes $2P5C$

Figure A.9 – Évaluation (ϕ) moyenne pour les problèmes $2P7C$ Figure A.10 – Évaluation (ϕ) moyenne pour les problèmes $3P3C$

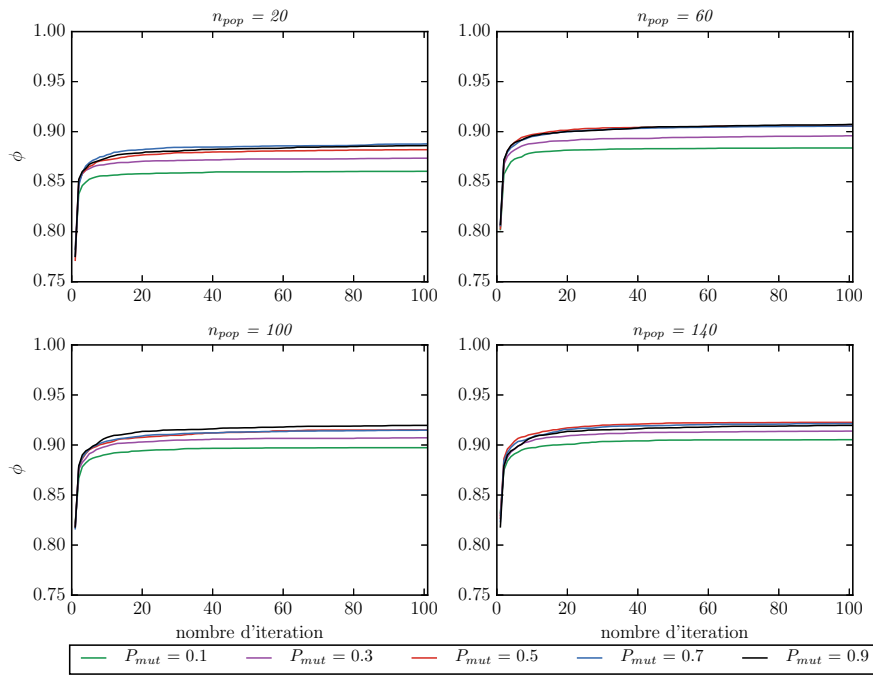


Figure A.11 – Évaluation (ϕ) moyenne pour les problèmes $3P5C$

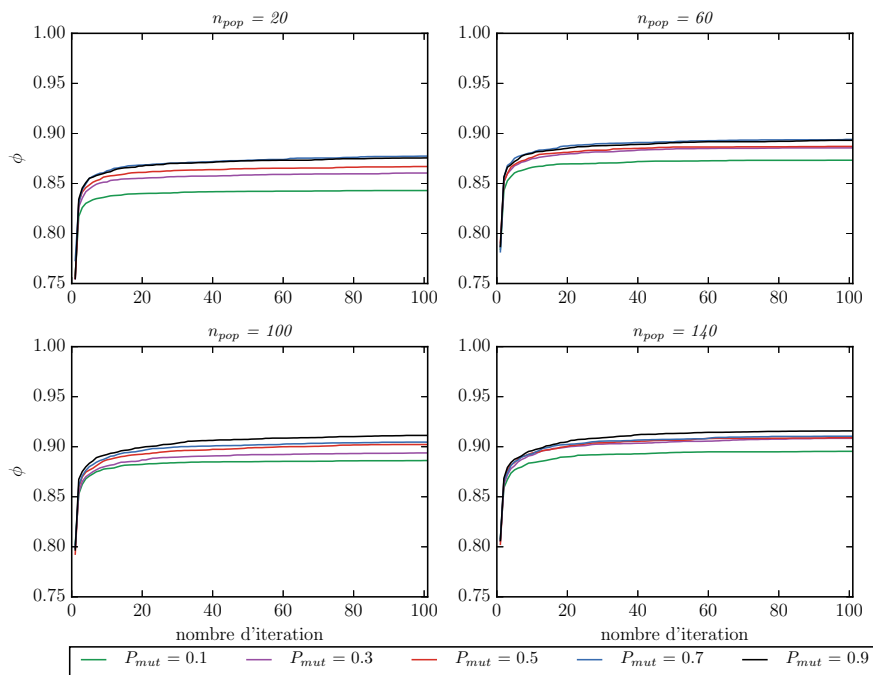


Figure A.12 – Évaluation (ϕ) moyenne pour les problèmes $3P7C$

A.3 Résultats des test de la configuration 3

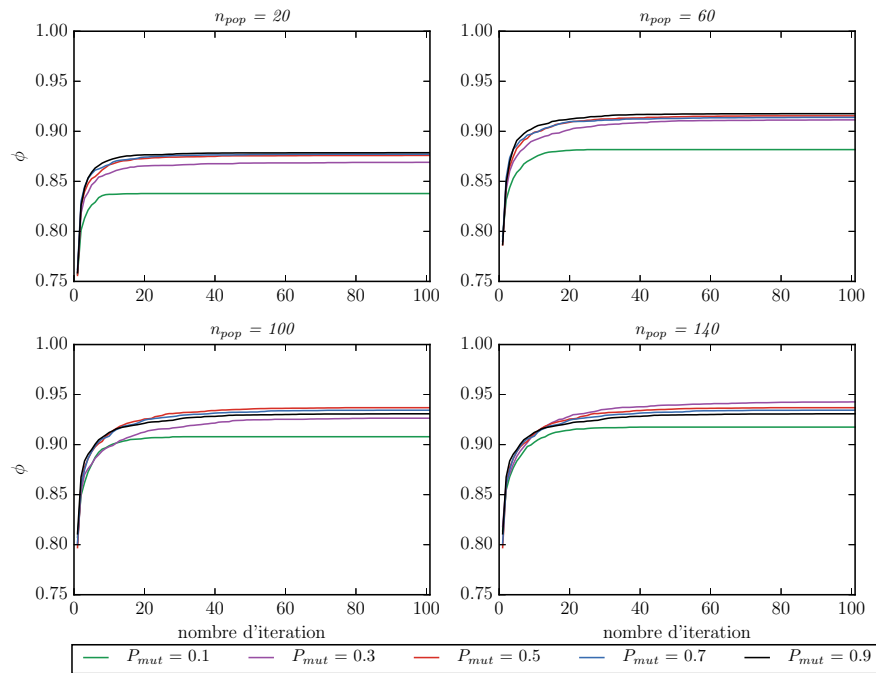


Figure A.13 – Évaluation (ϕ) moyenne pour les problèmes $2P3C$

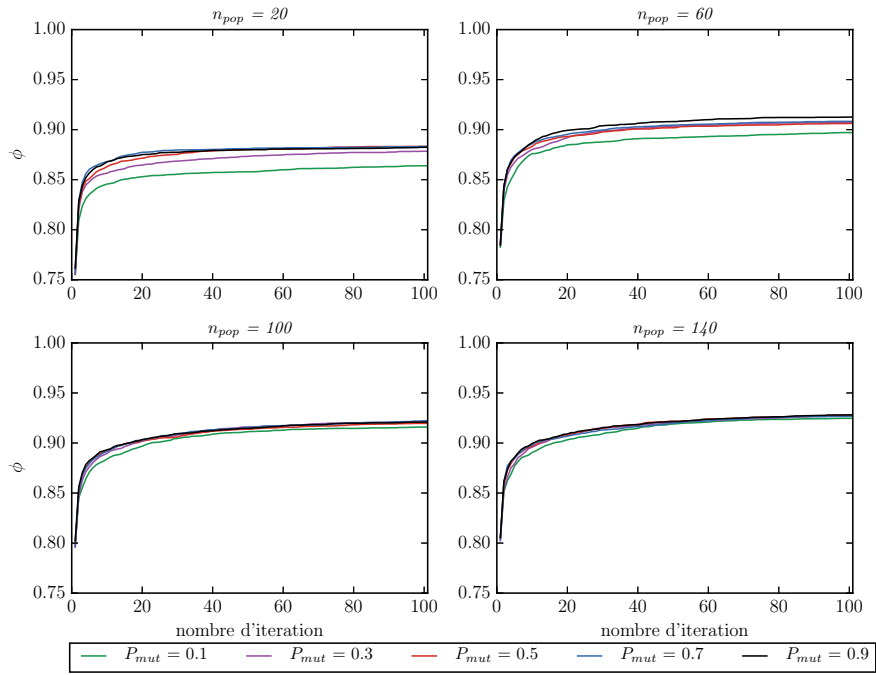


Figure A.14 – Évaluation (ϕ) moyenne pour les problèmes $2P5C$

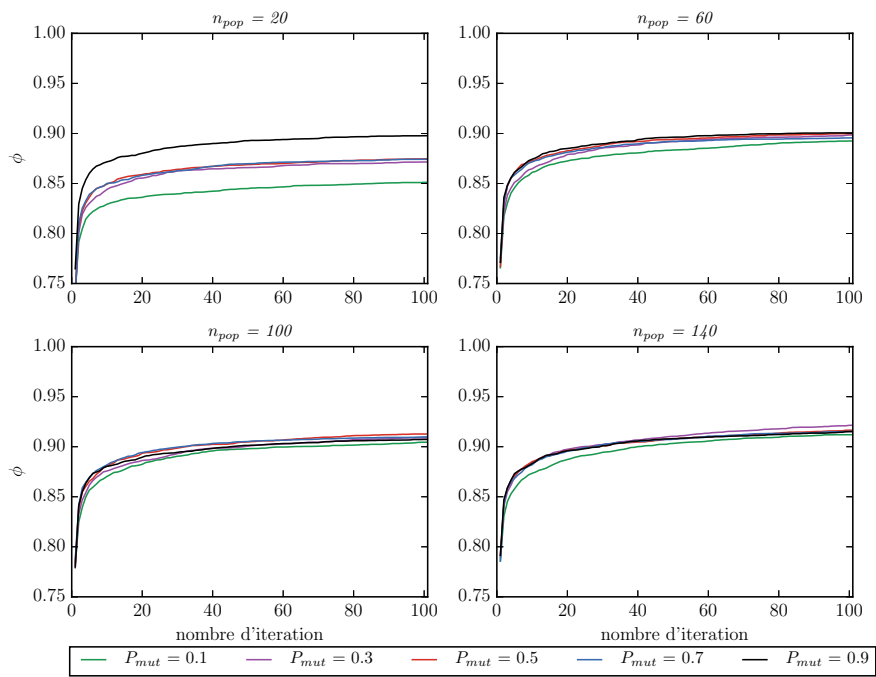
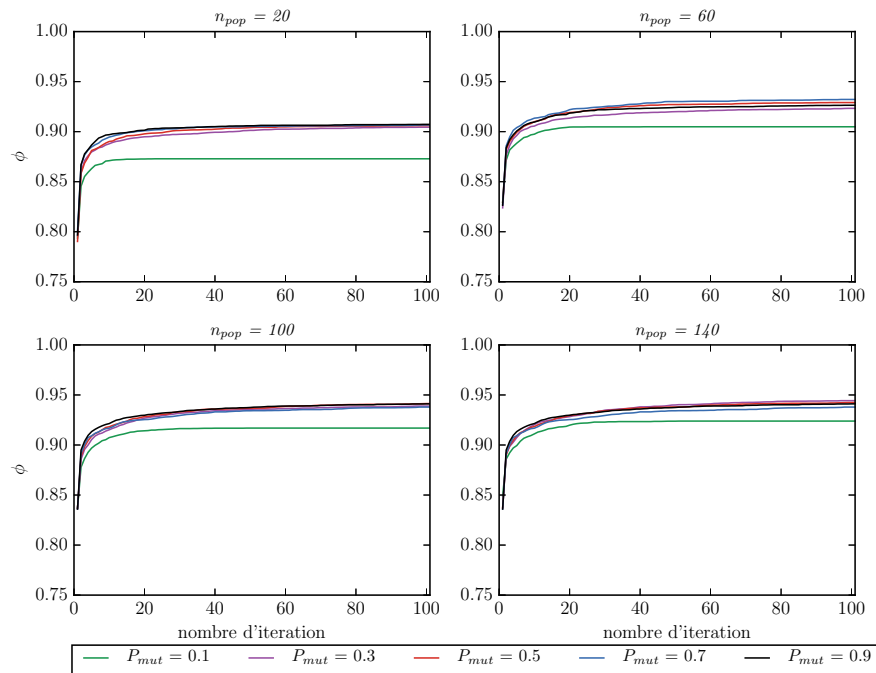
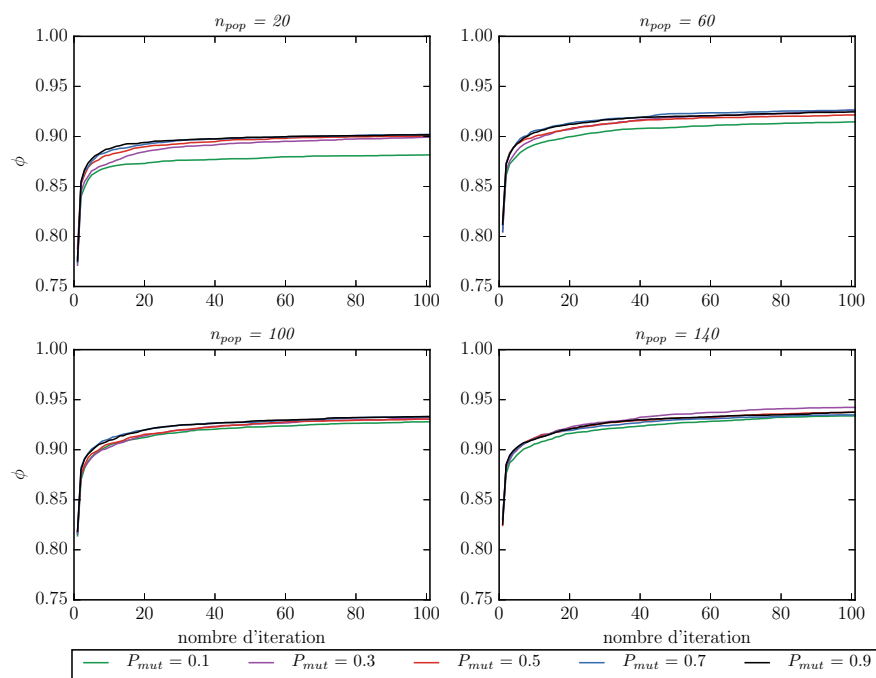
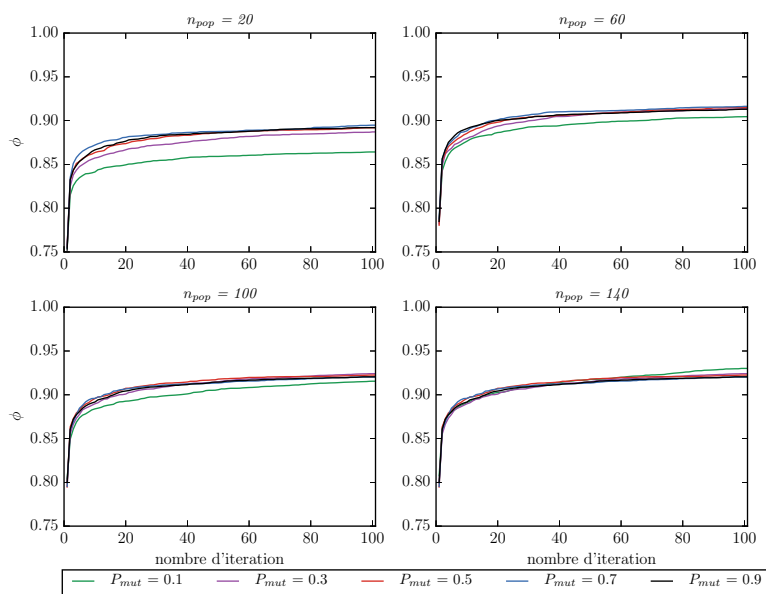
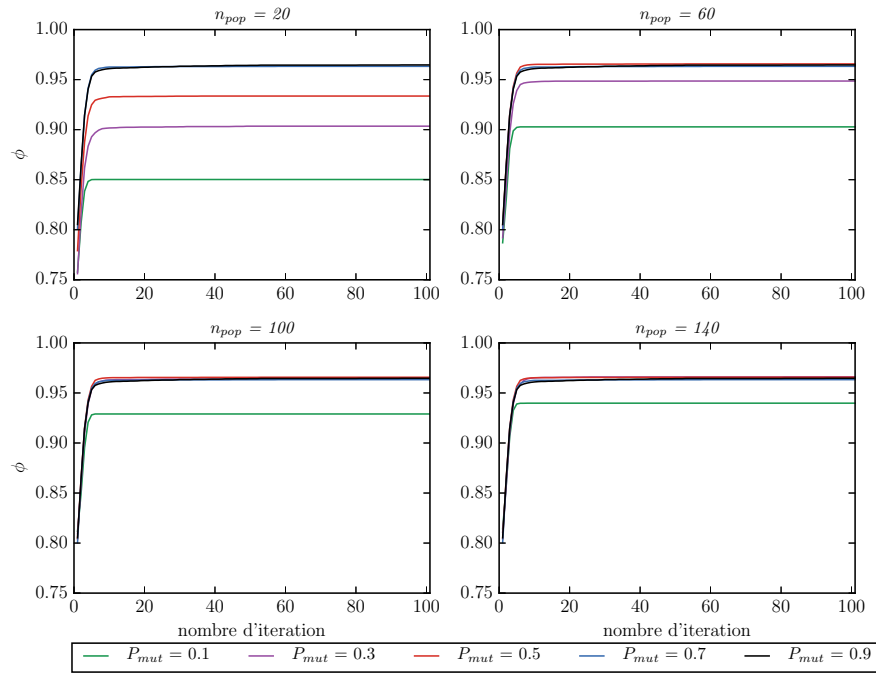
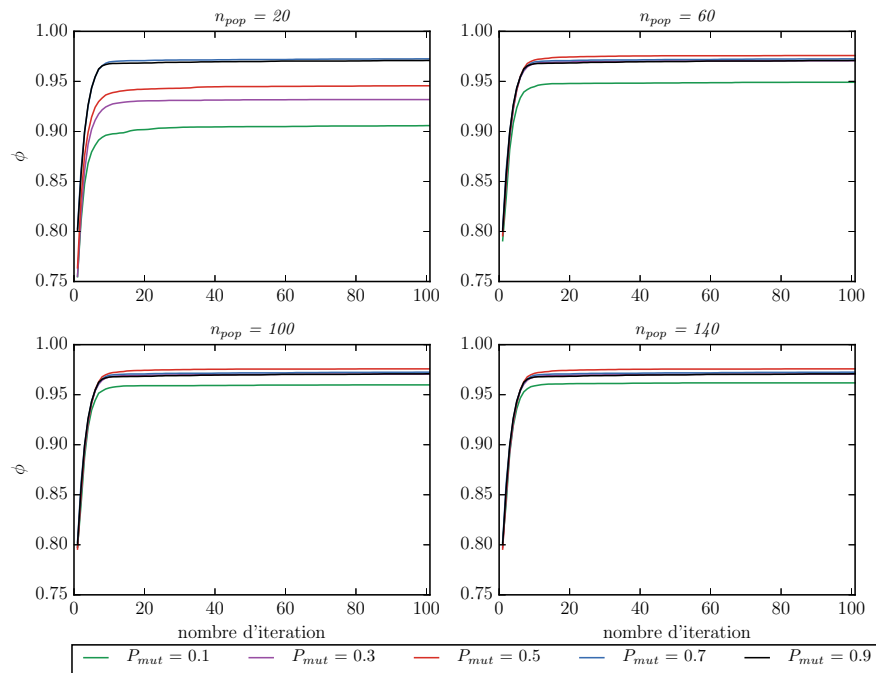


Figure A.15 – Évaluation (ϕ) moyenne pour les problèmes $2P7C$

Figure A.16 – Évaluation (ϕ) moyenne pour les problèmes $3P3C$ Figure A.17 – Évaluation (ϕ) moyenne pour les problèmes $3P5C$

Figure A.18 – Évaluation (ϕ) moyenne pour les problèmes $3P7C$

A.4 Résultats des test de la configuration 4

Figure A.19 – Évaluation (ϕ) moyenne pour les problèmes $2P3C$ Figure A.20 – Évaluation (ϕ) moyenne pour les problèmes $2P5C$

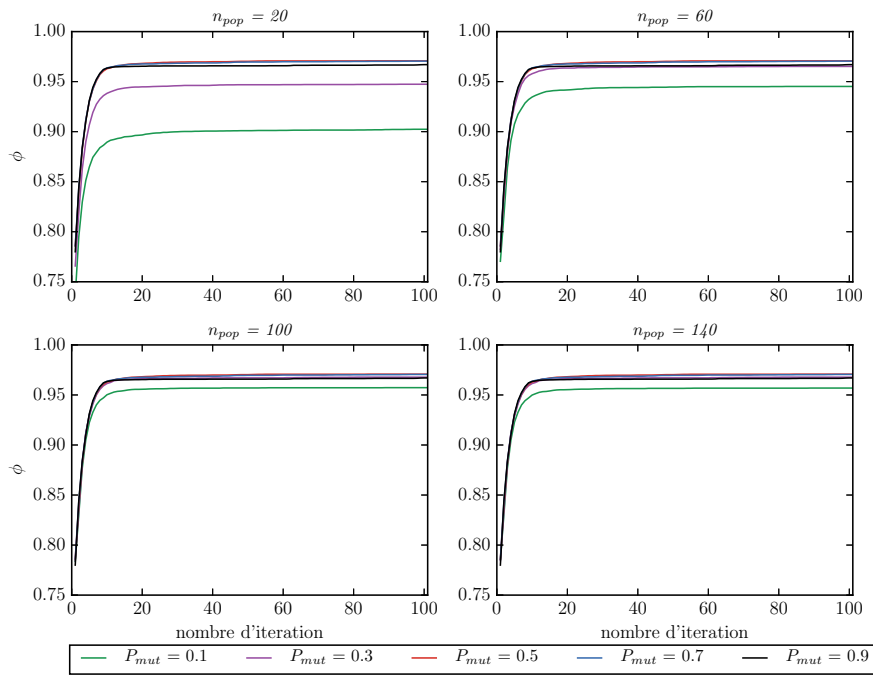


Figure A.21 – Évaluation (ϕ) moyenne pour les problèmes $2P7C$

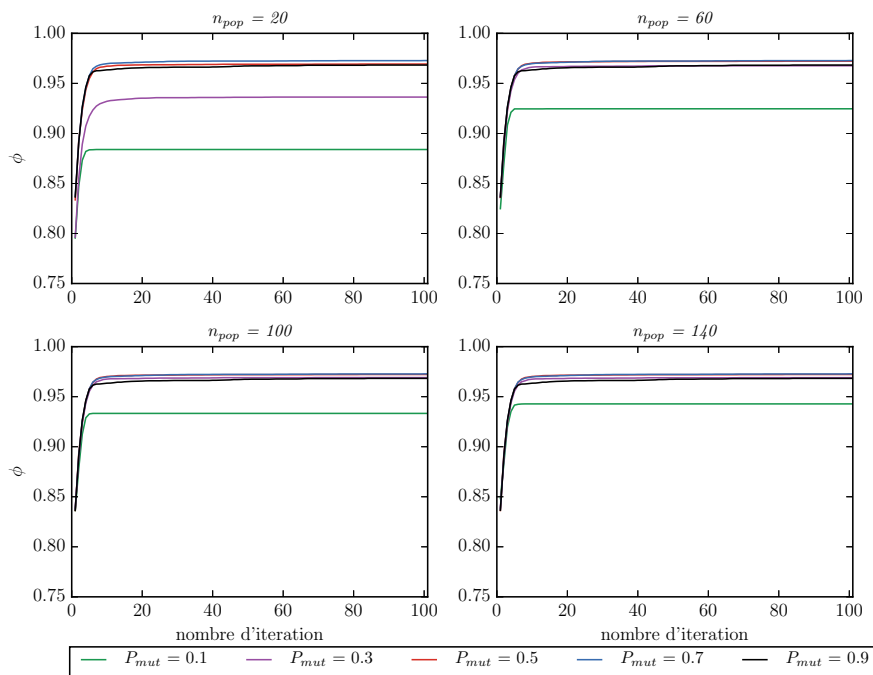
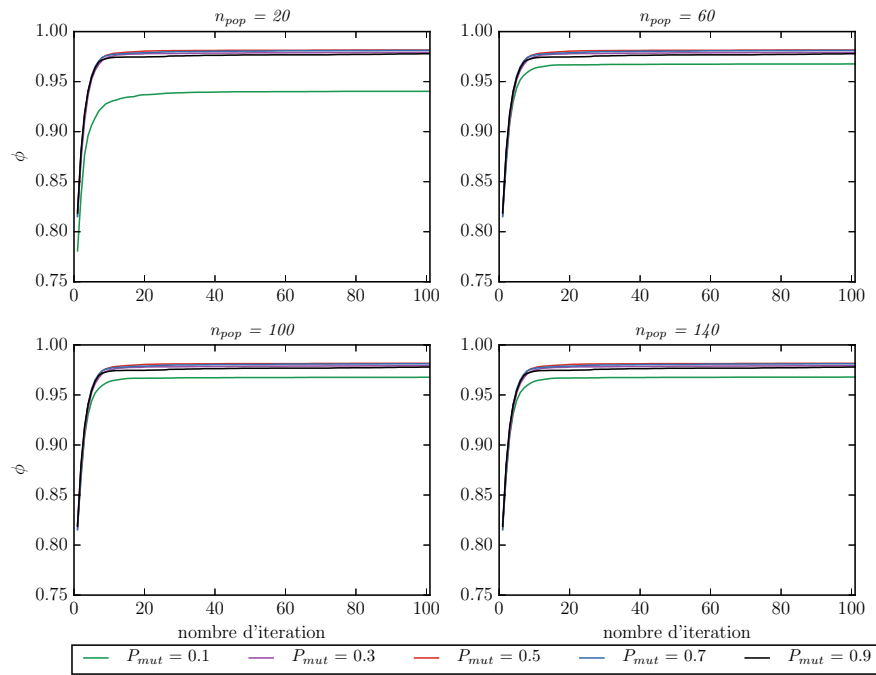
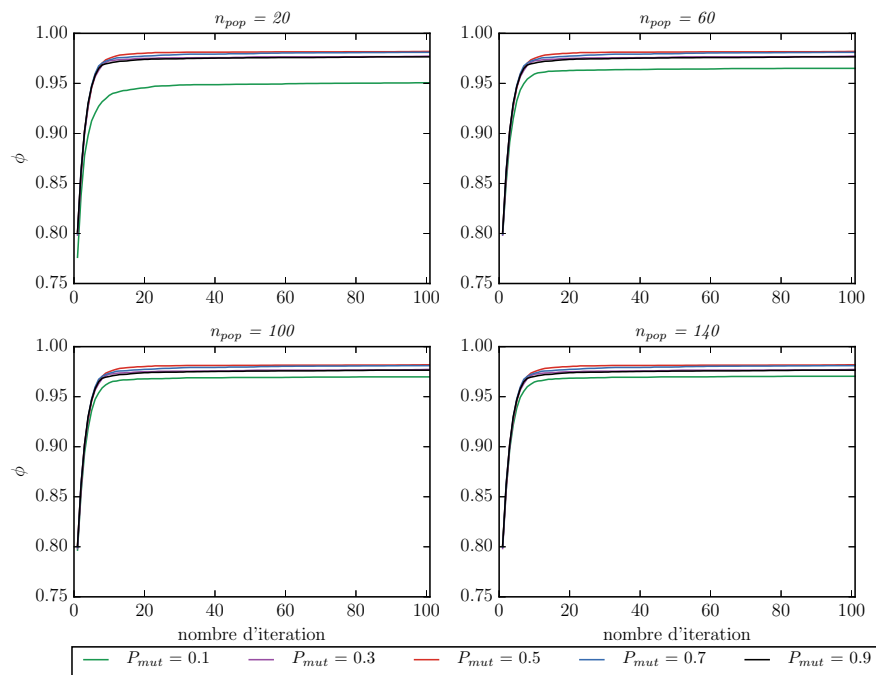


Figure A.22 – Évaluation (ϕ) moyenne pour les problèmes $3P3C$

Figure A.23 – Évaluation (ϕ) moyenne pour les problèmes $3P5C$ Figure A.24 – Évaluation (ϕ) moyenne pour les problèmes $3P7C$

Annexe B

Les programmes linéaires mixte en nombres entiers

<i>Paramètres :</i>	
$A^*, Pr, In, M, k, \sigma, \gamma$	(1)
$a_j \in [0, 1]$	$\forall a \in A^*, \forall j \in M$ (2)
pC_j^h	$\forall h \in 1..k, \forall j \in M$ (3)
$wC_j \in [0, 1]$	$\forall j \in M$ (4)
$\alpha \in [0, 1]$	(5)
$\alpha_I^h \in [0, 1]$	$\forall h \in 1..k$ (6)
$\beta \in [0, 1]$	(7)
M	(8)
<i>Variables :</i>	
$w_j \in [0, 1]$	$\forall j \in M$ (9)
p_j^h	$\forall h \in 1..k, \forall j \in M$ (10)
$\delta_j(a, p^h) \in \{0, 1\}$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M$ (11)
$\omega_j(a, p^h) \in [0, 1]$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M$ (12)
$s(a, b, p^h) \in \{0, 1\}$	$\forall (a, b) \in Pr, \forall h \in 0..k$ (13)
$absP_j^h \geq 0$	$\forall h \in 1..k, \forall j \in M$ (14)
$absW_j \geq 0$	$\forall j \in M$ (15)
$absS1(p^h) \in \{0, 1\}$	$\forall h \in 0..k$ (16)
$absS2(w_j) \in \{0, 1\}$	$\forall j \in M$ (17)
<i>Objectif :</i>	
$minimiser : \alpha * \sum_{h=1}^k \alpha_I^h \sum_{j \in M} absP_j^h / m + \beta * \sum_{j \in M} absW_j / m$	(18)
<i>Contraintes :</i>	
<i>s.t.</i> $\sum_{j \in M} w_j = 1$	(19)
$w_j \geq \gamma$	$\forall j \in M$ (20)
$p_j^1 \geq 0$	$\forall j \in M$ (21)
$1 \geq p_j^k$	$\forall j \in M$ (22)
$p_j^{h+1} \geq p_j^h$	$\forall h \in 1..k-1, \forall j \in M$ (23)
$a_j - p_j^h \geq \delta_j(a, p^h) - 1$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M$ (24)
$\delta_j(a, p^h) \geq a_j - p_j^h + \gamma$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M$ (25)
$w_j \geq \omega_j(a, p^h)$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M$ (26)
$\omega_j(a, p^h) \geq 0$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M$ (27)
$\delta_j(a, p^h) \geq \omega_j(a, p^h)$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M$ (28)
$\omega_j(a, p^h) \geq \delta_j(a, p^h) + w_j - 1$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M$ (29)
$s(a, b, p^0) = 1$	$\forall (a, b) \in Pr$ (30)
$s(a, b, p^k) = 0$	$\forall (a, b) \in Pr$ (31)
$\sum_{j \in M} \omega_j(a, p^{\sigma(h)}) \geq \sum_{j \in M} \omega_j(b, p^{\sigma(h)}) + \gamma - s(a, b, p^{\sigma(h)}) \cdot (1 + \gamma) - (1 - s(a, b, p^{\sigma(h-1)}))$	$\forall (a, b) \in Pr, \forall h \in 1..k$ (32)
$\sum_{j \in M} \omega_j(a, p^{\sigma(h)}) \geq \sum_{j \in M} \omega_j(b, p^{\sigma(h)}) - (1 - s(a, b, p^{\sigma(h)})) - (1 - s(a, b, p^{\sigma(h-1)}))$	$\forall (a, b) \in Pr, \forall h \in 1..k$ (33)
$\sum_{j \in M} \omega_j(b, p^{\sigma(h)}) + (1 - s(a, b, p^{\sigma(h)})) + (1 - s(a, b, p^{\sigma(h-1)})) \geq \sum_{j \in M} \omega_j(a, p^{\sigma(h)})$	$\forall (a, b) \in Pr, \forall h \in 1..k$ (34)
$\sum_{j \in M} \omega_j(a, p^{\sigma(h)}) = \sum_{j \in M} \omega_j(b, p^{\sigma(h)})$	$\forall (a, b) \in In, \forall h \in 1..k$ (35)
$absP_j^h \geq p_j^h - pC_j^h - M * absS1(p^h)$	$\forall h \in 1..k, \forall j \in M$ (36)
$absP_j^h \geq -p_j^h + pC_j^h - M * (absS1(p^h) - 1)$	$\forall h \in 1..k, \forall j \in M$ (37)
$absW_j \geq w_j - wC_j - M * absS2(w_j)$	$\forall h \in 1..k, \forall j \in M$ (38)
$absW_j \geq -w_j + wC_j - M * (absS2(w_j) - 1)$	$\forall j \in M$ (39)

Tableau B.1 – MIP pour inférer l'emodèle SRMP le plus proche

<i>Paramètres :</i>	
$A^*, Pr, In, M, k, \sigma, \gamma$	(1)
$a_j \in [0, 1]$	$\forall a \in A^*, \forall j \in M$ (2)
$\alpha \in [0, 1]$	(3)
$\alpha_I^h \in [0, 1]$	$\forall h \in 1..k$ (4)
$\beta \in [0, 1]$	(5)
M	(6)
<i>Variables :</i>	
$w_{j,l} \in [0, 1]$	$\forall j \in M, \forall l \in 1, 2$ (7)
$p_{j,l}^h$	$\forall h \in 1..k, \forall j \in M, \forall l \in 1, 2$ (8)
$\delta_j(a, p_l^h) \in \{0, 1\}$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M, \forall l \in 1, 2$ (9)
$\omega_j(a, p_l^h) \in [0, 1]$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M, \forall l \in 1, 2$ (10)
$s(a, b, p_l^h) \in \{0, 1\}$	$\forall (a, b) \in Pr, \forall h \in 0..k, \forall l \in 1, 2$ (11)
$absP_j^h \geq 0$	$\forall h \in 1..k, \forall j \in M$ (12)
$absW_j \geq 0$	$\forall j \in M$ (13)
$absS1(p^h) \in \{0, 1\}$	$\forall h \in 0..k$ (14)
$absS2(w_j) \in \{0, 1\}$	$\forall j \in M$ (15)
<i>Objectif :</i>	
$maximize : \alpha * \sum_{h=1}^k \alpha_I^h \sum_{j \in \mathcal{M}} absP_j^h / m / k + \beta * \sum_{j \in \mathcal{M}} absW_j / m$	(16)
<i>Contraintes :</i>	
<i>s.t.</i> $\sum_{j \in M} w_{j,l} = 1$	$\forall l \in 1, 2$ (17)
$w_{j,l} \geq \gamma$	$\forall j \in M, \forall l \in 1, 2$ (18)
$p_{j,l}^1 \geq 0$	$\forall j \in M, \forall l \in 1, 2$ (19)
$1 \geq p_{j,l}^k$	$\forall j \in M, \forall l \in 1, 2$ (20)
$p_{j,l}^{h+1} \geq p_{j,l}^h$	$\forall h \in 1..k - 1, \forall j \in M, \forall l \in 1, 2$ (21)
$a_j - p_{j,l}^h \geq \delta_j(a, p_l^h) - 1$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M, \forall l \in 1, 2$ (22)
$\delta_j(a, p_j^h) \geq a_j - p_{j,l}^h + \gamma$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M, \forall l \in 1, 2$ (23)
$w_{j,l} \geq \omega_j(a, p_l^h)$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M, \forall l \in 1, 2$ (24)
$\omega_j(a, p_l^h) \geq 0$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M, \forall l \in 1, 2$ (25)
$\delta_j(a, p_l^h) \geq \omega_j(a, p_l^h)$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M, \forall l \in 1, 2$ (26)
$\omega_j(a, p_l^h) \geq \delta_j(a, p_l^h) + w_{j,l} - 1$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M, \forall l \in 1, 2$ (27)
$s(a, b, p_l^0) = 1$	$\forall (a, b) \in Pr, \forall l \in 1, 2$ (28)
$s(a, b, p_l^k) = 0$	$\forall (a, b) \in Pr, \forall l \in 1, 2$ (29)
$\sum_{j \in M} \omega_j(a, p_l^{\sigma(h)}) \geq \sum_{j \in M} \omega_j(b, p_l^{\sigma(h)}) + \gamma - s(a, b, p_l^{\sigma(h)}) \cdot (1 + \gamma) - (1 - s(a, b, p_l^{\sigma(h-1)}))$	$\forall (a, b) \in Pr, \forall h \in 1..k, \forall l \in 1, 2$ (30)
$\sum_{j \in M} \omega_j(a, p_l^{\sigma(h)}) \geq \sum_{j \in M} \omega_j(b, p_l^{\sigma(h)}) - (1 - s(a, b, p_l^{\sigma(h)})) - (1 - s(a, b, p_l^{\sigma(h-1)}))$	$\forall (a, b) \in Pr, \forall h \in 1..k, \forall l \in 1, 2$ (31)
$\sum_{j \in M} \omega_j(b, p_l^{\sigma(h)}) + (1 - s(a, b, p_l^{\sigma(h)})) + (1 - s(a, b, p_l^{\sigma(h-1)})) \geq \sum_{j \in M} \omega_j(a, p_l^{\sigma(h)})$	$\forall (a, b) \in Pr, \forall h \in 1..k, \forall l \in 1, 2$ (32)
$\sum_{j \in M} \omega_j(a, p_l^{\sigma(h)}) = \sum_{j \in M} \omega_j(b, p_l^{\sigma(h)})$	$\forall (a, b) \in In, \forall h \in 1..k$ (33)
$absP_j^h \geq p_j^h - pC_j^h - M * absS1(p^h)$	$\forall h \in 1..k, \forall j \in M$ (34)
$absP_j^h \geq -p_j^h + pC_j^h - M * (absS1(p^h) - 1)$	$\forall h \in 1..k, \forall j \in M$ (35)
$absW_j \geq w_j - wC_j - M * absS2(w_j)$	$\forall h \in 1..k, \forall j \in M$ (36)
$absW_j \geq -w_j + wC_j - M * (absS2(w_j) - 1)$	$\forall j \in M$ (37)

Tableau B.2 – MIP pour inférer les modèles SRMP d'extrémité

<i>Paramètres :</i>	
$A^*, \mathcal{P}r, \mathcal{I}n, M, k, \sigma, \gamma$	(1)
$a_j \in [0, 1]$	$\forall a \in A^*, \forall j \in M$ (2)
$pMin_j^h$	$\forall h \in 1..k, \forall j \in M$ (3)
$wMin_j \in [0, 1]$	$\forall j \in M$ (4)
$pMax_j^h$	$\forall h \in 1..k, \forall j \in M$ (5)
$wMax_j \in [0, 1]$	$\forall j \in M$ (6)
$\alpha \in [0, 1]$	(7)
$\alpha_I^h \in [0, 1]$	$\forall h \in 1..k$ (8)
$\beta \in [0, 1]$	(9)
M	(10)
<i>Variabes :</i>	
$w_j \in [0, 1]$	$\forall j \in M$ (11)
p_j^h	$\forall h \in 1..k, \forall j \in M$ (12)
$\delta_j(a, p^h) \in \{0, 1\}$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M$ (13)
$\omega_j(a, p^h) \in [0, 1]$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M$ (14)
$s(a, b, p^h) \in \{0, 1\}$	$\forall (a, b) \in \mathcal{P}r, \forall h \in 0..k$ (15)
$absPMin_j^h \geq 0$	$\forall h \in 1..k, \forall j \in M$ (16)
$absWMin_j \geq 0$	$\forall j \in M$ (17)
$absSMin1(p^h) \in \{0, 1\}$	$\forall h \in 0..k$ (18)
$absSMin2(w_j) \in \{0, 1\}$	$\forall j \in M$ (19)
$absPMax_j^h \geq 0$	$\forall h \in 1..k, \forall j \in M$ (20)
$absWMax_j \geq 0$	$\forall j \in M$ (21)
$absSMax1(p^h) \in \{0, 1\}$	$\forall h \in 0..k$ (22)
$absSMax2(w_j) \in \{0, 1\}$	$\forall j \in M$ (23)
$\Delta \geq 0$	(24)
$V \in \{0, 1\}$	(25)
<i>Objectif :</i>	
<i>maximize :</i>	
$(\alpha * \sum_{h=1}^k \alpha_I^h \sum_{j \in \mathcal{M}} absPMin_j^h / m / k + \beta * \sum_{j \in \mathcal{M}} absWMin_j / m) +$	(26)
$(\alpha * \sum_{h=1}^k \alpha_I^h \sum_{j \in \mathcal{M}} absPMax_j^h / m / k + \beta * \sum_{j \in \mathcal{M}} absWMax_j / m) + \Delta$	
<i>Contraintes :</i>	
<i>s.t.</i> $\sum_{j \in M} w_j = 1$	(26)
$w_j \geq \gamma$	$\forall j \in M$ (27)
$p_j^1 \geq 0$	$\forall j \in M$ (28)
$1 \geq p_j^k$	$\forall j \in M$ (29)
$p_j^{h+1} \geq p_j^h$	$\forall h \in 1..k-1, \forall j \in M$ (30)
$a_j - p_j^h \geq \delta_j(a, p^h) - 1$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M$ (31)
$\delta_j(a, p^h) \geq a_j - p_j^h + \gamma$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M$ (32)
$w_j \geq \omega_j(a, p^h)$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M$ (33)
$\omega_j(a, p^h) \geq 0$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M$ (34)
$\delta_j(a, p^h) \geq \omega_j(a, p^h)$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M$ (35)
$\omega_j(a, p^h) \geq \delta_j(a, p^h) + w_j - 1$	$\forall a \in A^*, \forall h \in 1..k, \forall j \in M$ (36)
$s(a, b, p^0) = 1$	$\forall (a, b) \in \mathcal{P}r$ (37)
$s(a, b, p^k) = 0$	$\forall (a, b) \in \mathcal{P}r$ (38)
$\sum_{j \in M} \omega_j(a, p^{\sigma(h)}) \geq \sum_{j \in M} \omega_j(b, p^{\sigma(h)}) + \gamma - s(a, b, p^{\sigma(h)}) \cdot (1 + \gamma) - (1 - s(a, b, p^{\sigma(h-1)}))$	$\forall (a, b) \in \mathcal{P}r, \forall h \in 1..k$ (39)
$\sum_{j \in M} \omega_j(a, p^{\sigma(h)}) \geq \sum_{j \in M} \omega_j(b, p^{\sigma(h)}) - (1 - s(a, b, p^{\sigma(h)})) - (1 - s(a, b, p^{\sigma(h-1)}))$	$\forall (a, b) \in \mathcal{P}r, \forall h \in 1..k$ (40)
$\sum_{j \in M} \omega_j(b, p^{\sigma(h)}) + (1 - s(a, b, p^{\sigma(h)})) + (1 - s(a, b, p^{\sigma(h-1)})) \geq \sum_{j \in M} \omega_j(a, p^{\sigma(h)})$	$\forall (a, b) \in \mathcal{P}r, \forall h \in 1..k$ (41)
$\sum_{j \in M} \omega_j(a, p^{\sigma(h)}) = \sum_{j \in M} \omega_j(b, p^{\sigma(h)})$	$\forall (a, b) \in \mathcal{I}n, \forall h \in 1..k$ (42)
$(\alpha * \sum_{h=1}^k \alpha_I^h \sum_{j \in \mathcal{M}} absPMin_j^h / m / k + \beta * \sum_{j \in \mathcal{M}} absWMin_j / m) -$	
$(\alpha * \sum_{h=1}^k \alpha_I^h \sum_{j \in \mathcal{M}} absPMax_j^h / m / k + \beta * \sum_{j \in \mathcal{M}} absWMax_j / m) +$	(43)
$M * V \geq \Delta$	
$-(\alpha * \sum_{h=1}^k \alpha_I^h \sum_{j \in \mathcal{M}} absPMin_j^h / m / k + \beta * \sum_{j \in \mathcal{M}} absWMin_j / m) +$	
$(\alpha * \sum_{h=1}^k \alpha_I^h \sum_{j \in \mathcal{M}} absPMax_j^h / m / k + \beta * \sum_{j \in \mathcal{M}} absWMax_j / m) +$	(44)
$M * (1 - V) \geq \Delta$	
$(\alpha * \sum_{h=1}^k \alpha_I^h \sum_{j \in \mathcal{M}} absPMin_j^h / m / k + \beta * \sum_{j \in \mathcal{M}} absWMin_j / m) -$	
$(\alpha * \sum_{h=1}^k \alpha_I^h \sum_{j \in \mathcal{M}} absPMax_j^h / m / k + \beta * \sum_{j \in \mathcal{M}} absWMax_j / m) +$	(45)
$M * V \leq \Delta$	
$-(\alpha * \sum_{h=1}^k \alpha_I^h \sum_{j \in \mathcal{M}} absPMin_j^h / m / k + \beta * \sum_{j \in \mathcal{M}} absWMin_j / m) +$	
$(\alpha * \sum_{h=1}^k \alpha_I^h \sum_{j \in \mathcal{M}} absPMax_j^h / m / k + \beta * \sum_{j \in \mathcal{M}} absWMax_j / m) +$	(46)
$M * (1 - V) \leq \Delta$	

Tableau B.3 – MIP pour inférer un modèle SRMP central

Annexe C

Temps d'exécution de l'algorithme génétique

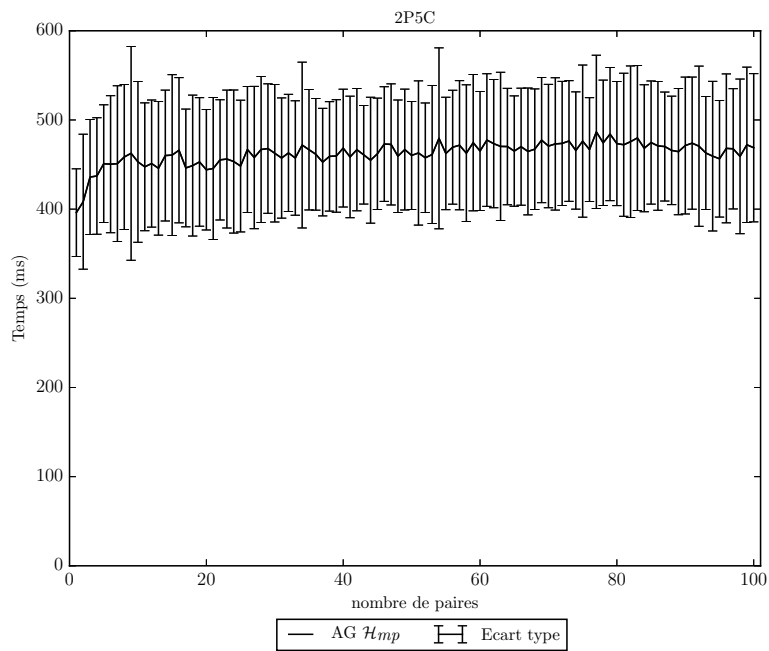
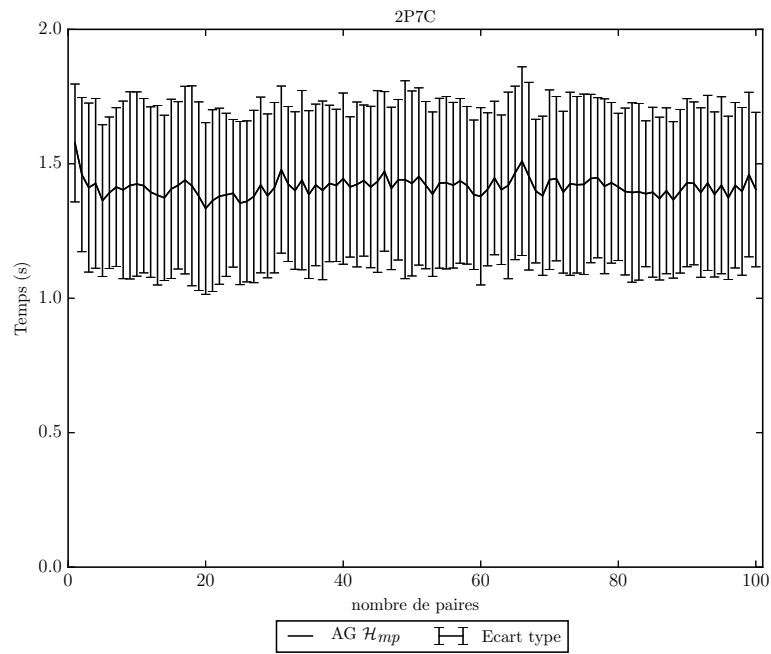
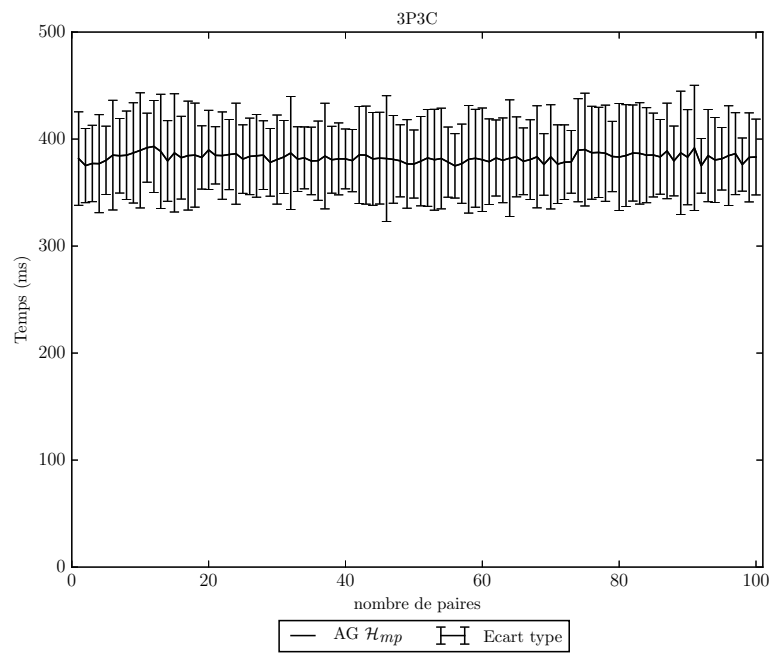


Figure C.1 – Le temps d'exécution moyen et l'écart type pour les problèmes $2P\ 5C$

Figure C.2 – Le temps d'exécution moyen et l'écart type pour les problèmes $2P\ 7C$ Figure C.3 – Le temps d'exécution moyen et l'écart type pour les problèmes $3P\ 3C$

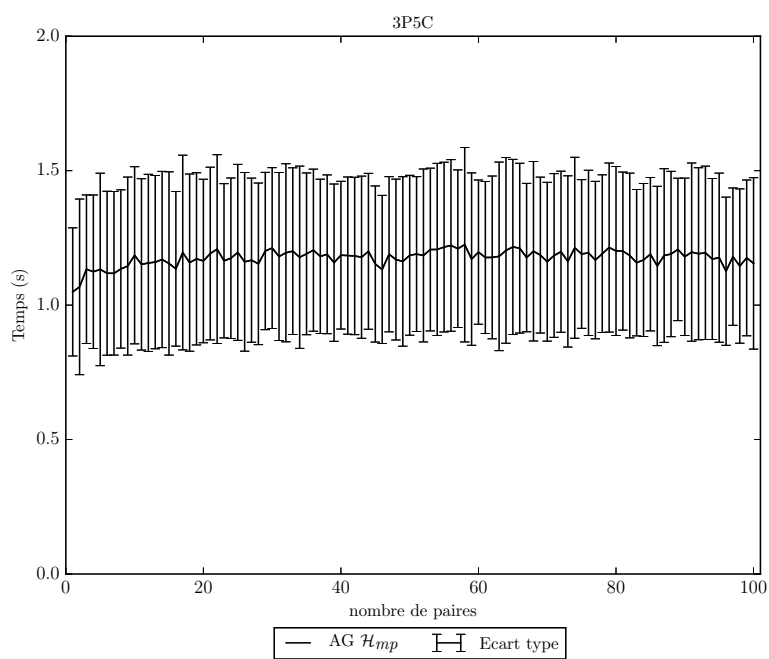


Figure C.4 – Le temps d'exécution moyen et l'écart type pour les problèmes $3P\ 5C$

Titre : Intégration des préférences d'un opérateur dans les décisions d'un drone autonome et élicitation incrémentale de ces préférences

Mots clés : drone autonome, aide multi-critère à la décision, élicitation incrémentale des préférences, confiance de l'opérateur

Résumé : Un drone totalement autonome est un aéronef sans pilote humain à bord. Il est donc capable d'accomplir une mission sans l'intervention d'un opérateur humain et de prendre des décisions de façon totalement autonome. Cela sous-entend que l'opérateur au sol doit avoir une confiance élevée dans les décisions prises par le drone. L'objectif principal de cette thèse est donc de proposer un moteur de décisions à embarquer dans le drone autonome qui garantit un niveau de confiance élevé de l'opérateur dans la capacité du drone à prendre les "bonnes" décisions. Pour cela nous proposons un moteur de décisions multi-niveaux composé de deux niveaux de décisions principaux. Le premier permet de surveiller l'état du drone et de son environnement pour détecter les événements qui peuvent perturber la réalisation de la mission et déclencher la prise

de décision du second niveau. Celui-ci une fois déclenché permet de choisir une action de haut niveau (atterrir, continuer, ...) la mieux adaptée à la situation courante parmi un ensemble d'actions possibles. Ce moteur intègre aussi les préférences d'un opérateur en utilisant des modèles d'Aide Multi-Critère à la Décision. Ces modèles nécessitent une phase en amont de la mission, où les préférences de l'opérateur sont élicitées, avant d'être intégrées dans le drone. Pour réduire l'effort cognitif de l'opérateur pendant cette phase, nous proposons un processus d'élicitation incrémental pendant lequel les questions soumises à l'opérateur sont déduites des réponses précédentes. Cela nous permet de déterminer un modèle représentant fidèlement ses préférences, tout en minimisant le nombre de questions.

Title: Integration of an operator's preferences into the decisions of an unmanned aerial vehicle and incremental elicitation of these preferences

Keywords : unmanned aerial vehicle, multi-criteria decision aiding, incremental preference elicitation, operator confidence

Abstract: A fully autonomous unmanned aerial vehicle (UAV) is an aircraft without a human pilot on board. It is consequently able to accomplish a mission without the intervention of a human operator and to make decisions in a totally autonomous way. This implies that the ground operator must have a high level of confidence in the decisions made by the UAV. The main objective of this thesis is therefore to propose a decision engine to be embedded in the autonomous UAV that guarantees a high level of operator confidence in the UAV's ability to make the "right" decisions. For this purpose, we propose a multi-level decision engine composed of two main decision levels. The first one monitors the state of the UAV and its environment to detect events that can disrupt

the mission's execution and trigger the second level. Once triggered, it allows to choose a high-level action (landing, continuing,...) best adapted to the current situation from a set of possible actions. This engine also integrates the operator's preferences by using Multi-Criteria Decision Aiding models. They require a preliminary phase before the mission, where the operator's preferences are elicited, before being integrated into the UAV. To reduce the operator's effort during this phase, we propose an incremental elicitation process during which the questions submitted to the operator are deduced from the previous answers. This allows us to determine a model that accurately represents his or her preferences, while minimizing the number of questions.