

N° d'ORDRE :

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET
DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE
« HOUARI BOUMEDIENE »
FACULTE DE MATHÉMATIQUES



MEMOIRE

Présenté pour l'obtention du diplôme de MAGISTER
EN : Recherche Opérationnelle
Spécialité : Mathématiques Discrètes et Optimisation
Par : Asma Skoudarli

Thème

**Prise en compte de l'incertitude dans
l'optimisation combinatoire multi-objectif**

Soutenu le 06/07/2011 à 09h30, devant le jury composé de :

Mr M. MOULAÏ,	Professeur ,	U.S.T.H.B.	Président
Mr M. AÏDER,	Professeur,	U.S.T.H.B.	Directeur du mémoire
Mr D. CHAABANE	Maître de Conférences classe A,	U.S.T.H.B.	Examineur
Mme C. ADICHE	Maître de Conférences classe B,	U.S.T.H.B.	Invitée

Remerciements

Tout d'abord, je remercie ALLAH le tout puissant, de m'avoir donnée le courage, la force et la volonté pour la réalisation de ce travail.

Je tiens très sincèrement à exprimer toute ma profonde gratitude et ma reconnaissance à mon directeur de mémoire, Monsieur Méziane Aïder, Professeur à l'U.S.T.H.B., pour l'honneur qu'il m'a fait en assurant la direction du présent mémoire. Je le remercie pour sa disponibilité et pour la pertinence de ses conseils avec lesquelles il a su me guider durant mes travaux de recherche.

Je tiens également à remercier chaleureusement Monsieur El-Ghazali Talbi, Professeur à l'université de Lille 1, pour la confiance qu'il m'a témoignée en m'accueillant dans son équipe de recherche DOLPHIN ainsi que son énorme soutien scientifique.

Je remercie vivement Monsieur Mustapha Moulai, Professeur à l'U.S.T.H.B., qui a bien voulu me faire honneur en acceptant de prendre en charge la présidence du jury.

Je remercie également Monsieur Djamel Chaabane, Maître de Conférences à l'U.S.T.H.B., et Madame Chahrazed Adiche, Maître de Conférences à l'U.S.T.H.B., pour l'honneur qu'ils m'ont fait en acceptant de faire partie du jury.

Je tiens à remercier plus personnellement mes très chers parents pour leurs sacrifices, leurs générosité, leurs éducation et leurs précieux conseils. Je tiens à remercier mes chers frères et soeurs pour leurs soutiens et leurs encouragements tout au long de ce travail.

Table des matières

Introduction générale	1
1 Optimisation multi-objectif	4
1.1 Introduction	4
1.2 Définitions et concepts de base	5
1.3 Difficulté d'un PMO	9
1.4 Approches de résolution	10
1.4.1 Point de vue du décideur	11
1.4.2 Point de vue algorithmique	12
1.5 Métaheuristiques pour l'optimisation multi-objectif	13
1.5.1 Affectation d'une valeur de fitness	13
1.5.2 Préservation de la diversité	17
1.5.3 Élitisme	19
1.6 Recherche locale multi-objectif	20
1.7 Problèmes d'optimisation multi-objectif	21
1.8 Conclusion	21
2 Problèmes de tournées de véhicules	22
2.1 Introduction	22
2.2 Problèmes de tournées de véhicules	23
2.3 Domaines d'application	25
2.4 VRP déterministe	25
2.5 Variantes du VRP	28
2.6 Approches de résolution du VRP et ses variantes	30

2.7	VRP du mono-objectif au multi-objectif	33
2.8	Approches de résolution du VRP multi-objectif	35
2.9	VRP incertain	38
2.10	Conclusion	38
3	Modélisation d'une information imparfaite	39
3.1	Introduction	39
3.2	Concepts de base	39
3.3	Modélisation des imperfections	41
3.3.1	Théorie des probabilités	41
3.3.2	Théorie des sous-ensembles flous (Fuzzy set)	43
3.3.3	Théorie des possibilités	47
3.3.4	Théorie des fonctions de croyance	49
3.4	Comparaison entre les différentes théories	53
3.5	Relations de dominance et fonctions de croyance	55
3.6	Conclusion	59
4	Approche Algorithmique	60
4.1	Introduction	60
4.2	Algorithme DMLS	60
4.2.1	Description des composants	62
4.2.2	Algorithmes existants	64
4.3	Application au VRP avec demandes incertaines	65
4.4	Implémentation sous ParadisEO	68
4.4.1	Présentation de la plateforme ParadisEO	68
4.4.2	Présentation du module ParadisEO-MOEO	69
4.5	Analyse expérimentale	70
4.6	Design expérimental	71
4.7	Résultats	73
4.8	Conclusion	78
	Conclusion générale	79
	Bibliographie	80

Table des figures

1.1	Espace décisionnel et espace des objectifs.	5
1.2	Solutions dominées et non dominées dans l'espace objectif.	6
1.3	Exemples de frontières Pareto.	7
1.4	Solutions supportées et non supportées.	8
1.5	Bornes du front Pareto.	8
1.6	Méthode d'agrégation.	14
1.7	Méthode ϵ -contrainte.	14
1.8	Schéma de fonctionnement de VEGA.	15
1.9	Méthode lexicographique dans le cas bi-objectif.	15
1.10	Stratégies de dominance.	16
1.11	Stratégies de diversité.	17
1.12	Classification des approches d'optimisation multi-objectif.	19
1.13	Recherche locale.	20
2.1	Vehicle Routing Problem (VRP)	23
2.2	Différents objectifs pour le VRP	33
3.1	Quelques éléments caractéristiques de A	45
3.2	Fonction d'appartenance exclusive T_{grand}	45
3.3	Fonction d'appartenance floue T_{grand}	46
3.4	Fonction de croyance.	51
3.5	Fonction de plausibilité.	52
3.6	Croyance(Bel) et Plausibilité (Pl)	52

3.7	Position de la théorie des possibilités et la théorie des probabilités dans le cadre de la théorie de l'évidence d'après Bouchon-Meunier [7]	54
3.8	Valeurs de f (a) certaines et (b) incertaines.	55
3.9	Dominance certaine ($y \prec_c y'$).	56
3.10	Dominance incertaine ($y \prec_{uc} y'$).	57
3.11	Indifférence ($y \parallel y'$).	57
3.12	Dominance faible ($y \prec_w y'$).	58
3.13	Dominance forte ($y \prec_s y'$).	58
3.14	Indifférence ($y \parallel y'$).	59
4.1	Étapes d'un algorithme DMLS.	61
4.2	Exemple de modification de la route due au caractère incertain des demandes.	67
4.3	Modules de la plateforme ParadisEO.	69
4.4	Opérateur de voisinage 2-Opt.	71
4.5	Indicateur hypervolume.	76

Liste des tableaux

2.1	Problèmes de tournées de véhicules multi-objectif de la littérature.	36
2.2	Problèmes de tournées de véhicules multi-objectif de la littérature.	37
4.1	Instance C101.	72
4.2	Résultats du cas déterministe.	73
4.3	Résultats du cas incertain.	73
4.4	Comparaison des fronts selon l'indicateur d'hypervolume.	77

Introduction générale

Ce mémoire s'inscrit dans le domaine de l'optimisation combinatoire multi-objectif et plus particulièrement dans l'étude et la résolution de problèmes difficiles issus de la logistique tels que les problèmes de tournées de véhicules.

L'étude et la résolution de problèmes comportant plusieurs objectifs est l'objet de l'optimisation multi-objectif. Ce domaine possède ses racines dans les travaux de Edgeworth [15] et de Pareto [64] dans le cadre d'études d'économie au 19^{ème} siècle. Cependant, ce domaine a connu un intérêt croissant ces deux dernières décennies avec l'apparition des métaheuristiques. De manière formelle, un problème d'optimisation multi-objectif est un problème de la forme "opt" $f(x) = (f_1(x), f_2(x), \dots, f_n(x))$ tel que x soit une solution réalisable. Les composants de f sont les différents objectifs à optimiser. La notion de solution optimale n'a pas de sens dans la mesure où les objectifs sont souvent conflictuels. La solution d'un problème multi-objectif n'est pas une unique solution mais un ensemble de solutions dites *Pareto optimales*. Ces solutions sont celles pour lesquelles l'amélioration d'un des objectifs entraîne systématiquement la détérioration de la qualité d'au moins un autre objectif. L'union de ses solutions forme l'ensemble *Pareto optimal*, et l'image de cet ensemble dans l'espace des objectifs est appelé le *front Pareto*. L'une des questions difficiles est donc liée à l'identification de l'ensemble Pareto optimal, ou d'une approximation de celui-ci pour des problèmes complexes.

Actuellement, l'optimisation multi-objectif est appliquée dans de nombreux domaines académiques et industriels. En particulier, les problèmes de logistique qui font partie des principaux problèmes étudiés en recherche opérationnelle. Dans le cadre de ce mémoire, nous nous intéressons à un problème d'optimisation multi-objectif issu des problèmes de trans-

port en général, et de problèmes de routage en particulier. Il s'agit du problème de tournées de véhicules.

Le problème de tournées de véhicules (*Vehicle Routing Problem VRP*) sert à modéliser de nombreux problèmes réels. Il consiste en l'élaboration de routes permettant de visiter des clients à partir d'un ou de plusieurs dépôts. L'objectif est de créer ces routes en ne prenant en compte qu'une unique fonction objectif qui vise le plus souvent à minimiser les coûts de transport. Or, la plupart des problèmes rencontrés dans la logistique sont de nature multi-objectif. Les variantes sont multiples et, selon la complexité du problème réel, diverses contraintes peuvent être introduites.

Par ailleurs, une grande partie des problèmes d'optimisation réels est soumise à des incertitudes devant être prises en compte. En dépit de l'importance de l'optimisation multi-objectif et de l'optimisation sous incertitude, très peu de travaux consacrés à l'optimisation multi-objectif sous incertitude existent à ce jour. En outre, bien qu'elles soient généralement adaptables au cas combinatoire, les études existantes n'ont été expérimentées que sur des problèmes continus ou sur des fonctions de test. Or, il n'est pas évident que les performances soient identiques pour des problèmes combinatoires. Par exemple, les problèmes de tournées sont des problèmes combinatoires habituellement étudiés sous une forme mono-critère déterministe. Pourtant, il existe de nombreux problèmes de tournées multi-objectif présentant des incertitudes dans leurs description.

En pratique, résoudre ce type de problèmes de façon exacte est souvent impossible en raison de la difficulté du problème, de l'aspect multi-objectif ou de l'incertitude des données. Ainsi, des métaheuristiques telles que les algorithmes évolutionnaires ont été utilisées pour résoudre ce type de problèmes. Cependant, en comparant le nombre considérable d'algorithmes évolutionnaires multi-objectif existants dans la littérature, la quantité d'approches de recherche locale multi-objectif est extrêmement faible. Ces dernières offrent pourtant une alternative intéressante aux algorithmes évolutionnaires classiques, et manipulent souvent un nombre relativement réduit de paramètres. Elles se basent généralement sur l'amélioration itérée d'une solution unique, et doivent dès lors être adaptées aux besoins de l'optimisation multi-objectif, et donc à la recherche d'un ensemble de solutions.

Dans notre travail, nous nous intéressons aux algorithmes de recherche locale basés sur une relation de dominance et pour la prise en compte de l'incertitude nous proposons d'utiliser de nouvelles relations de dominance entre des vecteurs d'intervalles au lieu de vecteurs précis.

Ce document est organisé comme suit :

Le premier chapitre est consacré aux définitions et concepts de base liés à l'optimisation multi-objectif et aux différentes approches de résolution des problèmes multi-objectif.

Dans le deuxième chapitre, nous présentons les problèmes de tournées de véhicules en général et les problèmes multi-objectif en particulier. Après avoir énuméré les principales caractéristiques des problèmes de tournées, un état de l'art sur les méthodes de résolution de ces problèmes est présenté.

Le troisième chapitre est consacré à l'environnement incertain et aux principales théories permettant de représenter des connaissances imparfaites. Nous présentons plus particulièrement la théorie des fonctions de croyance que nous utilisons ainsi que les nouvelles relations de dominance proposées.

Quant au quatrième chapitre, il se focalise autour de l'approche algorithmique et de son application au problème de tournées de véhicules de permutation bi-objectif avec demandes incertaines.

Le mémoire s'achève par une conclusion générale sur l'ensemble du travail réalisé et des perspectives de recherche induites par les résultats obtenus.

1

Optimisation multi-objectif

1.1 Introduction

Les problèmes d'optimisation rencontrés en pratique sont rarement mono-objectif. Il y a généralement plusieurs critères contradictoires à optimiser simultanément. Dans le cas de l'optimisation mono-objectif, la solution optimale est facilement définie ; il en est autrement dans le cas de plusieurs objectifs. Au lieu d'une solution unique, le résultat est généralement un ensemble de solutions, qui se distingue par différents compromis réalisés entre les objectifs.

L'étude et la résolution, même approchée, de problèmes multi-objectif présentent de nombreux défis. Pour ce faire, plusieurs métaheuristiques ont été développées pour résoudre ce type de problèmes. De même, de nombreux mécanismes et stratégies ont été proposées pour améliorer la qualité de ces métaheuristiques.

Dans ce chapitre, nous présentons dans un premier temps un ensemble de définitions liées aux problèmes d'optimisation multi-objectif. Par la suite, nous exposons une classification des différentes approches de résolution de ce type de problèmes.

1.2 Définitions et concepts de base

Cette section couvre les principales définitions et notations liées à l'optimisation multi-objectif, telles que la dominance de Pareto, l'optimalité Pareto, et le front Pareto [16, 84]. Sans perte de généralité, nous supposons par la suite que toutes les fonctions objectif sont à minimiser.

1.1.1 Problème multi-objectif

Définition 1.1 *Un problème d'optimisation multi-objectif PMO peut être défini de la manière suivante :*

$$(PMO) \begin{cases} \text{“min”} & f(x) = (f_1(x), \dots, f_n(x)) \\ & x \in S \end{cases}$$

où n est le nombre d'objectifs ($n \geq 2$), $x = (x_1, x_2, \dots, x_k) \in S$ est un vecteur représentant les variables de décision et S est l'ensemble des solutions réalisables dans l'espace *décisionnel*. Dans le cas combinatoire, S est un ensemble discret. A chaque solution $x \in S$ est associé un vecteur objectif $y \in Y$ sur la base d'un vecteur de fonctions $f : S \rightarrow Y$ tel que $y = (y_1, y_2, \dots, y_n) = f(x) = (f_1(x), f_2(x), \dots, f_n(x))$, où $Y = f(S)$ représente l'ensemble des points réalisables de l'espace *objectif* (FIG. 1.1). On note ce problème $PMO(f, S)$.

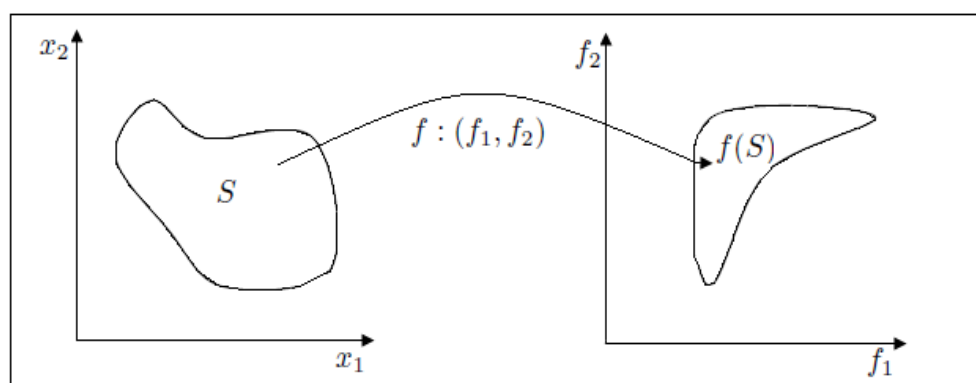


FIG. 1.1 – Espace décisionnel et espace des objectifs.

Dans le cas d'un problème d'optimisation mono-objectif, il existe un ordre total parmi les solutions réalisables, ce n'est plus le cas pour un problème d'optimisation multi-objectif. Il faut donc utiliser une autre notion d'optimalité. La plus communément admise est l'optimalité Pareto [64, 15].

1.1.2 Notions de dominance Pareto et d'optimalité

Définition 1.2 (Dominance Pareto) Soient f, f' deux vecteurs objectifs. On dit que f domine f' (et on note $f \prec_p f'$) si et seulement si :

$$\forall i \in \{1, 2, \dots, n\} : f_i \leq f'_i \quad \wedge \quad \exists j \in \{1, 2, \dots, n\} : f_j < f'_j.$$

Si f domine f' , alors f est au moins aussi bon que f' sur tous les objectifs et meilleure que lui sur au moins un objectif.

Définition 1.3 (Pareto optimalité) Une solution $x^* \in S$ est Pareto optimale ou non-dominée si et seulement si :

$$\forall x \in S, f(x) \not\prec_p f(x^*).$$

Le terme *Pareto optimalité* est aussi connu sous *efficacité* ou *non infériorité* (FIG. 1.2). Ces solutions Pareto optimales forment l'ensemble *Pareto optimal*, noté P^* . L'image de cet ensemble dans l'espace objectif est le *front Pareto*, et sera noté PF^* .

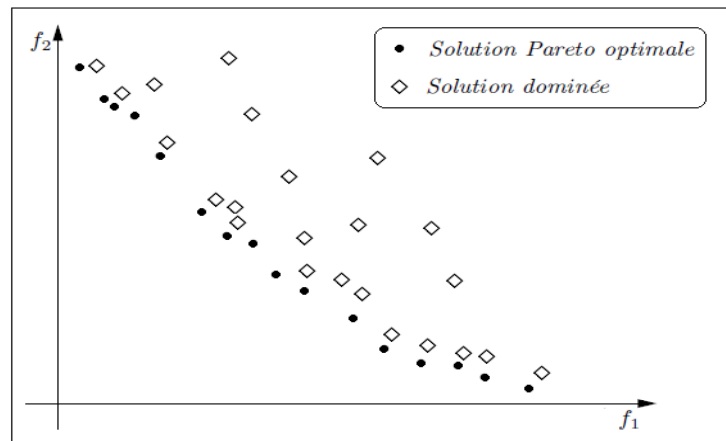


FIG. 1.2 – Solutions dominées et non dominées dans l'espace objectif.

Définition 1.4 (Ensemble Pareto optimal) *Étant donné un $PMO(f, S)$, l'ensemble Pareto optimal P^* est défini comme suit :*

$$P^* = \{x \in S / \nexists x' \in S, f(x') \prec_p f(x)\}.$$

Définition 1.5 (Front Pareto) *Étant donné un $PMO(f, S)$ et son ensemble Pareto optimal P^* , le front Pareto PF^* est défini comme suit :*

$$PF^* = \{f(x), x \in P^*\}.$$

La figure FIG. 1.3 présente, pour un problème bi-objectif, les quatre *frontières Pareto* en fonction du désir de l'utilisateur de minimiser ou maximiser les objectifs.

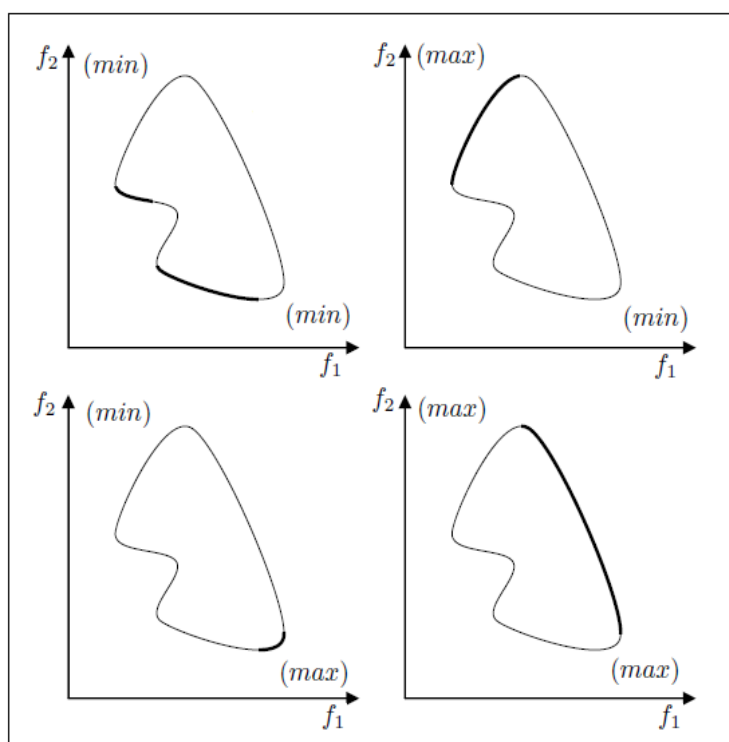


FIG. 1.3 – Exemples de frontières Pareto.

Le *front Pareto* comporte deux types de solutions : les *solutions supportées* et les *solutions non supportées*. L'image des premières solutions se trouvent sur l'enveloppe convexe de $f(S)$ (FIG. 1.4). L'image des secondes sont dominées par certaines combinaisons convexes de solutions supportées ; il s'agit de points à l'intérieur de l'enveloppe convexe de $f(S)$.

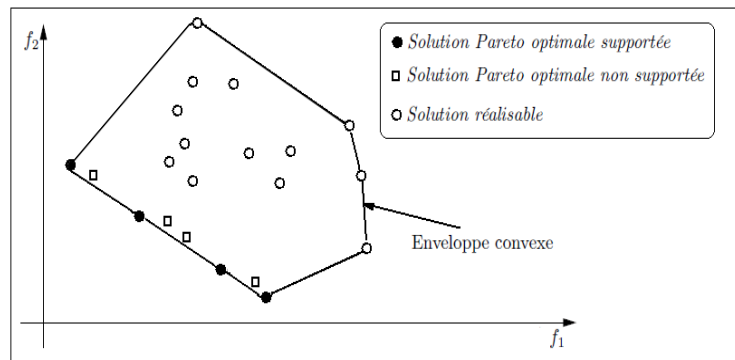


FIG. 1.4 – Solutions supportées et non supportées.

1.1.3 Bornes du front Pareto

Définition 1.6 (Point idéal) Le point idéal $y^* = (y_1^*, y_2^*, \dots, y_n^*)$ est le vecteur qui minimise chaque fonction objectif individuellement (FIG. 1.5) :

$$y_i^* = \min_{x \in S} f_i(x) \text{ pour } i \in \{1, 2, \dots, n\}.$$

Définition 1.7 (Point nadir) Le point nadir $y^n = (y_1^n, y_2^n, \dots, y_n^n)$ est le vecteur qui maximise chaque fonction objectif individuellement (FIG. 1.5) :

$$y_i^n = \max_{x \in S} f_i(x) \text{ pour } i \in \{1, 2, \dots, n\}.$$

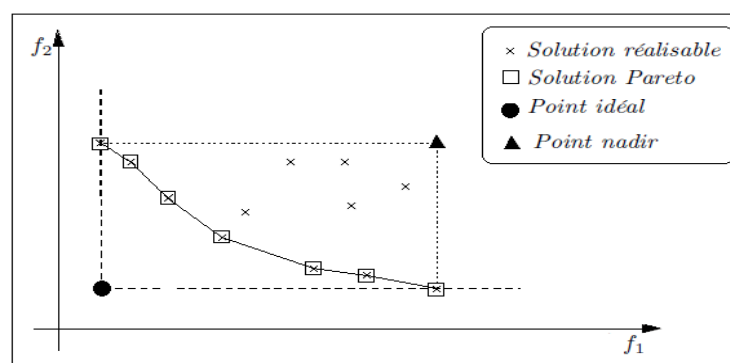


FIG. 1.5 – Bornes du front Pareto.

1.1.4 Autres notions de dominance

Définition 1.8 (Dominance forte) Soient f, f' deux vecteurs objectifs. On dit que f domine fortement f' (et on note $f \prec_s f'$) si et seulement si :

$$\forall i \in \{1, 2, \dots, n\} : f_i < f'_i.$$

Si f domine fortement f' , alors f est meilleur que f' sur tous les critères.

Définition 1.9 (Dominance faible) Soient f, f' deux vecteurs objectifs. On dit que f domine faiblement f' (et on note $f \prec_w f'$) si et seulement si :

$$\forall i \in \{1, 2, \dots, n\} : f_i \leq f'_i.$$

Définition 1.10 (ϵ -Dominance) Soient f, f' deux vecteurs objectifs. On dit que f ϵ -domine f' , avec $\epsilon > 1$ (et on note $f \prec_\epsilon f'$) si et seulement si :

$$\forall i \in \{1, 2, \dots, n\} : f_i \leq \epsilon f'_i.$$

1.3 Difficulté d'un PMO

Un processus d'optimisation multi-objectif doit résoudre les deux tâches suivantes :

- guider le processus de recherche vers la frontière de Pareto ;
- maintenir une diversité des solutions pour assurer une bonne répartition sur la frontière Pareto.

L'accomplissement de ces tâches est très délicat car les difficultés rencontrées dans un problème multi-objectif sont identiques à celles d'un problème mono-objectif. Elles sont amplifiées par la présence d'objectif dépendant les uns des autres.

Guider le processus de recherche vers la frontière Pareto

Le processus de recherche est souvent ralenti ou totalement dérouté par des fonctions possédant une des caractéristiques suivantes :

- *Multimodalité* : une fonction est dite multimodale si elle possède plusieurs optimaux globaux, dès lors, chaque optimum exerce sur les individus d'une population une attraction différente qui peut piéger la convergence de l'algorithme ;

- *Isolation d'un optimum* : il existe des problèmes dans lesquels un optimum peut être entouré de grandes zones pratiquement plates. Cet optimum se trouve alors isolé car l'espace de recherche qui l'entoure ne peut pas guider vers lui les individus de la population ;
- *Tromperie* : un problème est trompé lorsqu'il guide la convergence vers une zone non optimale de la fonction.

Maintenir la diversité sur le front Pareto

La difficulté à maintenir une bonne répartition des solutions sur la frontière de Pareto résulte principalement des caractéristiques suivantes :

- *Convexité ou non convexité* : certains problèmes ont une frontière de Pareto non convexe ;
- *Discontinuité* : si une frontière de Pareto est discontinue, nous retrouvons le même principe que pour une fonction multimodale. Les différentes parties de cette frontière vont exercer proportionnellement à leurs tailles, une attraction plus ou moins importante sur les individus d'une population, certaines d'entre elles pourront donc ne pas être découvertes ;
- *Non uniformité de la distribution* : les solutions sur la frontière de Pareto peuvent ne pas être réparties uniformément, la raison principale vient du choix des fonctions objectif, si une des fonctions objectif est multimodale, elle va influencer de manière très différente la répartition des solutions sur la frontière de Pareto.

1.4 Approches de résolution

Dans les différentes publications, nous rencontrons deux classifications différentes des approches de résolution de problèmes multi-objectif. La première classe adopte un point de vue de décideur, les méthodes sont classées en fonction de l'usage que l'on désire en faire. La deuxième classe est plus théorique, plus algorithmique. Cette section présente ces différentes approches de résolution des problèmes d'optimisation multi-objectif.

1.4.1 Point de vue du décideur

Résoudre un problème d'optimisation multi-objectif consiste à trouver l'ensemble Pareto optimal. Il est nécessaire de faire intervenir le décideur, pour le choix final de la solution ou des solutions à garder. Ainsi, il faut se poser la question sur le type de méthode d'optimisation à utiliser. En effet, on peut répartir les méthodes de résolution de problèmes multi-objectif en trois familles, en fonction du moment où intervient le décideur.

- *Les méthodes a priori (décideur \rightarrow recherche)* : Elles consistent souvent à combiner les différentes fonctions objectifs en une fonction d'utilité suivant les préférences du décideur. Dans ce cas le décideur est supposé connaître a priori le poids de chaque objectif afin de les combiner dans une fonction unique. Cela revient à résoudre un problème avec un seul objectif. Cependant, dans la plupart des cas, le décideur ne peut pas exprimer clairement sa fonction d'utilité, soit par manque d'expérience ou d'informations, soit parce que les différents objectifs sont exprimés dans des unités différentes.
- *Les méthodes a posteriori (recherche \rightarrow décideur)* : Le décideur prend sa décision d'après un ensemble de solutions calculé par un solveur. Dans ce cas la qualité de la décision dépend du choix de la méthode de résolution. Car celle-ci va devoir donner un ensemble de résultats le plus représentatif de l'espace des objectifs efficaces.
- *Les méthodes progressives ou interactives (décideur \leftrightarrow recherche)* : Dans ces méthodes, les processus de décision et d'optimisation sont alternés. Par moment, le décideur intervient de manière à modifier certaines variables ou contraintes afin de diriger le processus d'optimisation. Le décideur modifie ainsi interactivement le compromis entre ses préférences et les résultats.

1.4.2 Point de vue algorithmique

Les problèmes d'optimisation combinatoire multi-objectif sont pour la plupart NP-difficiles [16]. Pour leur résolution, des méthodes exactes, ainsi que des méthodes approchées ont été proposées. Concernant les méthodes exactes, plusieurs approches basées sur des procédures par séparation et évaluation “branch and bound” [76], sur l'algorithme A* [81] et la programmation dynamique [91] ont été proposées pour résoudre des problèmes bi-objectif de petites tailles. Pour les problèmes à plus de deux objectifs ou de grandes tailles, il n'existe pas de procédures exactes efficaces, étant données les difficultés dues simultanément à la complexité et à l'aspect multi-critère des problèmes. Des méthodes heuristiques sont nécessaires pour résoudre ce type de problèmes. Elles ne garantissent pas de trouver de manière exacte l'ensemble Pareto optimal, mais elles proposent en un temps de calcul raisonnable un ensemble de solutions potentiellement efficaces. Les méthodes heuristiques peuvent être divisées en deux classes : les algorithmes spécifiques à un problème donné qui utilisent des connaissances du domaine, et les algorithmes généraux applicables à une grande variété de problèmes d'optimisation multi-objectif. Dans ce travail, nous nous intéressons à la deuxième classe d'algorithmes, les *métaheuristiques*.

Deux types de métaheuristiques peuvent être distingués :

- Les *métaheuristiques à base de solution unique* (S-META) : telles que les algorithmes de recherche locale, de recherche tabou, de recuit simulé, etc. Elles manipulent et transforment une seule solution durant le processus de recherche ;
- Les *métaheuristiques à base de population* (P-META) : telles que les algorithmes évolutionnaires, les algorithmes à essaim de particules, etc. Dans les P-META les éléments d'un ensemble de solutions, appelé population, évoluent en parallèle.

Durant les deux dernières décennies, les métaheuristiques ont reçu un intérêt grandissant et certaines d'entre elles telles que : le recuit simulé [90], la recherche tabou [21] et les algorithmes évolutionnaires (algorithmes génétiques [80, 18], stratégies évolutionnistes [43]), ont montré leur efficacité dans de vastes domaines en résolvant de nombreux problèmes d'optimisation multi-objectif.

1.5 Métaheuristiques pour l'optimisation multi-objectif

Lors du développement d'une métaheuristique pour l'optimisation multi-objectif, deux critères contradictoires cités précédemment doivent être pris en compte :

Intensification : est la convergence vers la frontière Pareto ;

Diversification : est l'exploration de solutions diversifiées le long de la frontière Pareto.

Les S-META sont plutôt axées sur l'intensification. Les régions prometteuses sont explorées localement dans l'espoir de trouver de meilleures solutions. Les P-META sont généralement exploratoires et permettent une meilleure diversification de l'espace de recherche. En conséquence, les différences principales entre la conception d'une métaheuristique pour l'optimisation mono-objectif et d'une métaheuristique pour l'optimisation multi-objectif traitent de ces deux critères.

Ainsi, la conception de métaheuristiques pour l'optimisation multi-objectif se base sur les trois principaux composants de recherche suivants [84] :

- **Affectation d'une valeur de fitness** : une valeur scalaire, appelée fitness, est affectée à chaque vecteur objectif. Son rôle est de guider la recherche vers les solutions Pareto optimales pour une meilleure convergence.
- **Préservation de la diversité** : son rôle est de générer un ensemble diversifié de solutions Pareto optimales dans l'espace de décision et/ou objectif.
- **Élitisme** : a pour but la préservation et l'utilisation des solutions élites, c'est à dire les solutions non-dominées trouvées par l'algorithme.

1.5.1 Affectation d'une valeur de fitness

Dans le cas mono-objectif, la valeur de fitness affectée à une solution est généralement sa valeur objectif unidimensionnelle. Lors de la résolution d'un problème d'optimisation multi-objectif, la valeur de fitness d'une solution mesure sa qualité et vise à orienter la recherche vers les solutions Pareto optimales pour une meilleure convergence. Les stratégies d'affectation d'une valeur de fitness existantes sont classées en quatre familles : les approches scalaires, les approches basées sur un critère, les approches basées sur une relation de dominance (ou approches de Pareto), et les approches basées sur un indicateur.

Approches scalaires

Ces méthodes transforment un problème d'optimisation multi-objectif en un problème d'optimisation mono-objectif. Parmi ces méthodes on trouve :

Les méthodes d'*agrégation* : qui consistent à additionner les fonctions objectif en affectant à chacune d'elles un coefficient de poids (FIG. 1.6) ;

La méthode ϵ -*contrainte* : elle est basée sur l'optimisation d'un seul objectif f_i en considérant les autres objectifs f_j avec $j \neq i$ des contraintes et doivent être inférieures à une valeur ϵ_j (FIG. 1.7) ;

Les méthodes basées sur un *point de référence* sont d'autres exemples d'approches scalaires.

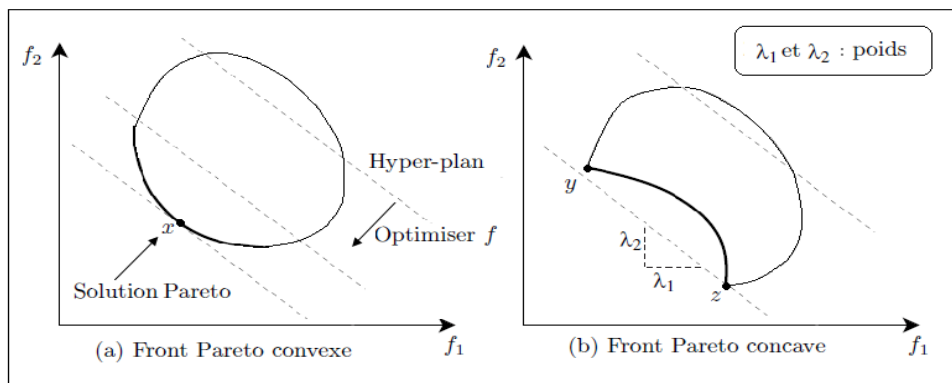


FIG. 1.6 – Méthode d'agrégation.

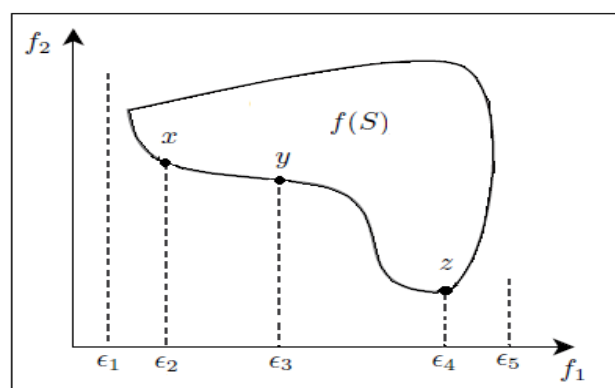


FIG. 1.7 – Méthode ϵ -contrainte.

Approches basées sur un critère

Ces approches possèdent un processus de recherche qui traite séparément les objectifs. L'exemple le plus classique est l'algorithme VEGA (*Vector Evaluated Genetic Algorithm*) proposé par Schaffer en 1985 [75], une sélection parallèle est effectuée afin de discerner les solutions de la population courante en fonction des valeurs sur une seule fonction objectif, indépendamment des autres (FIG. 1.8). Nous pouvons classer dans cette catégorie les méthodes *lexicographiques*, proposées par Fourman en 1985 [20], qui donnent un ordre de priorité sur les fonctions objectif à traiter. Ces fonctions sont alors traitées les unes à la suite des autres selon cet ordre (FIG. 1.9).

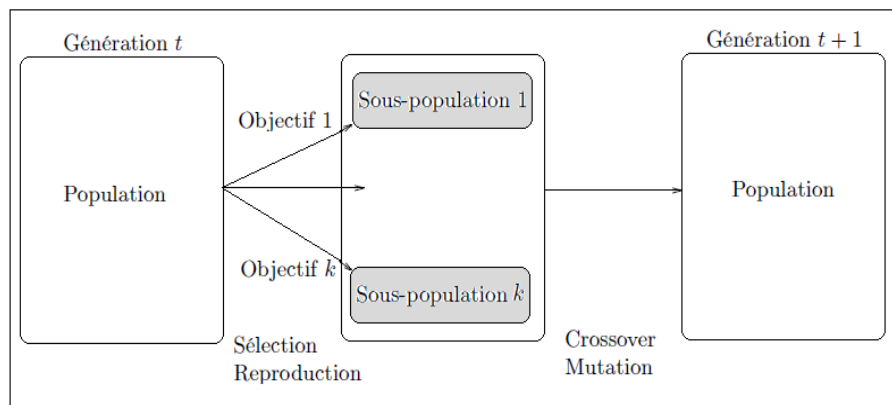


FIG. 1.8 – Schéma de fonctionnement de VEGA.

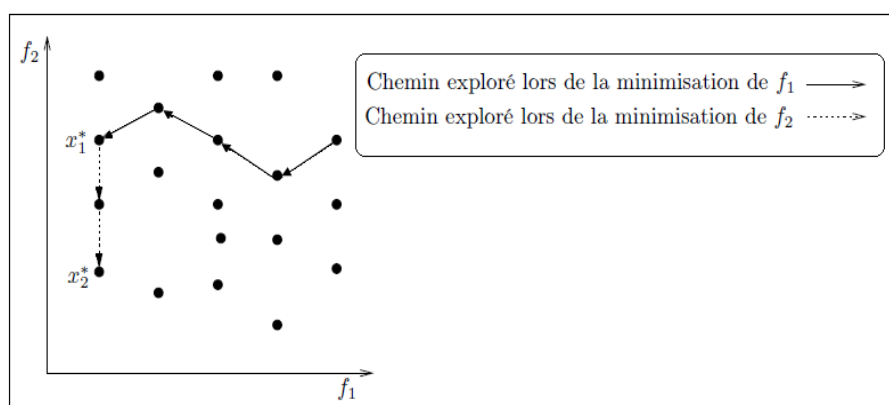


FIG. 1.9 – Méthode lexicographique dans le cas bi-objectif.

Approches basées sur une relation de dominance

Dans ces approches, une relation de dominance est définie pour classer les solutions de la population. Cette idée a été pour la première fois introduite par Goldberg en 1989 [24]. Il suggère d'utiliser la dominance de Pareto pour respecter l'intégralité de chaque fonction objectif. Trois stratégies principales peuvent être distinguées (FIG. 1.10) :

- (a) **Rang de dominance** (*dominance-rank*) : Cette technique consiste à calculer le nombre de solutions de la population courante qui dominent une solution donnée. Le rang de dominance a été pour la première fois utilisé par l'algorithme MOGA (*Multi-Objective Genetic Algorithm*) [19], où la valeur de fitness d'une solution est égale à son rang de dominance dans la population, incrémenté de un.
- (b) **Compte de dominance** (*dominance-count*) : Ici, la valeur de fitness d'une solution correspond au nombre d'individus qui sont dominés par cette solution. Cette approche peut être hybridée avec d'autres approches. Par exemple, dans SPEA (*Strength Pareto Evolutionary Algorithm*) [96] ou SPEA2 [95], compte de dominance est combiné au rang de dominance.
- (c) **Profondeur de dominance** (*dominance-depth*) : La population de solutions est classée en différents fronts. Une solution qui appartient à une classe ne domine aucun individu de cette même classe. Ainsi les solutions du premier front appartiennent au meilleur ensemble non-dominé ; celles du deuxième front au deuxième meilleur ensemble non-dominé, et ainsi de suite. Cette approche est utilisée dans les algorithmes NSGA (*Non-dominated Sorting Genetic Algorithm*) [80] et NSGA-II [12].

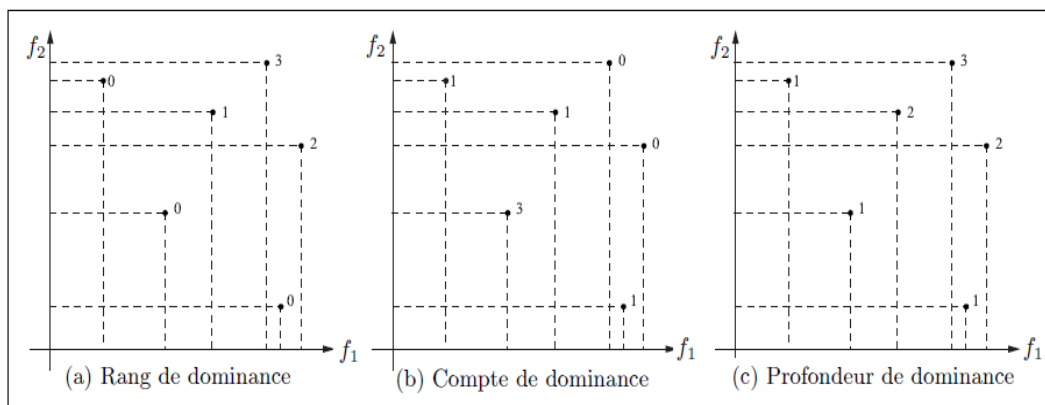


FIG. 1.10 – Stratégies de dominance.

Dans le cadre des approches basées sur une relation de dominance, la dominance Pareto est la plus couramment utilisée. Néanmoins, n'importe quelle autre relation peut être utilisée.

Approches basées sur un indicateur de qualité

Pour ce type d'approche, le but de l'optimisation est formulé à l'aide d'un indicateur de qualité, par exemple l'indicateur basé sur la notion d'*hypervolume* et l'indicateur basé sur la notion d' ϵ -dominance. Le choix de l'indicateur représente l'objectif général du processus de recherche. Ainsi des valeurs de fitness sont calculées en comparant les individus sur la base de cet indicateur de qualité. Parmi ces approches on trouvera l'algorithme IBEA (*Indicator-Based multiobjective Evolutionary Algorithm*) [94].

1.5.2 Préservation de la diversité

Les méthodes classiques d'affectation de la valeur de fitness présentées ci-dessus ont souvent tendance à converger de façon prématurée vers le front Pareto, ce qui ne garantit pas l'uniformité de l'ensemble trouvé. Afin d'éviter ce problème, un mécanisme de préservation de la diversité, basé sur une mesure de distance, doit être intégré dans la méthode de résolution afin de distribuer la population de manière uniforme le long du front Pareto. Trois stratégies principales peuvent être distinguées (FIG. 1.11) :

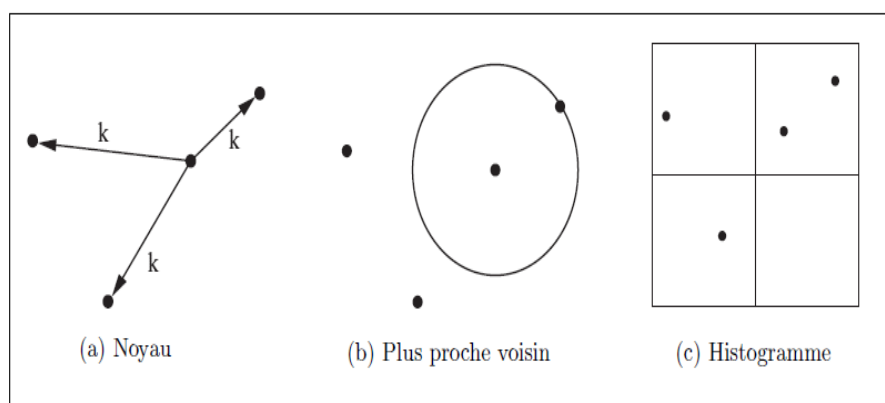


FIG. 1.11 – Stratégies de diversité.

(a) Méthodes par noyau

Ces méthodes définissent le voisinage d'une solution par rapport à une fonction noyau k prenant une mesure de distance comme argument. En pratique, pour chaque solution, on calcule la distance qui la sépare de chacun des autres éléments de la population. La fonction noyau k est alors appliquée à chaque distance, et les valeurs obtenues sont ensuite sommées. La somme des valeurs de la fonction k représente l'estimation de densité pour l'individu correspondant. Le *fitness sharing* [25] est la technique par noyau la plus populaire dans le cadre des algorithmes évolutionnaires. Elle est utilisée dans les algorithmes MOGA (*Multi-Objective Genetic algorithm*) [19], NPGA (*Niched Pareto Genetic Algorithm*) [28] et NSGA (*Non-dominates Sorting Genetic algorithm*) [80].

(b) Méthodes du plus proche voisin

Dans ces méthodes, la distance entre une solution et son k ème plus proche voisin est prise en compte pour estimer la densité d'une solution. Dans SPEA2 (*Strength Pareto Evolutionary Algorithm*) [95], l'estimateur de densité est basé sur l'inverse de cette distance. La préservation de la diversité opérée dans NSGA-II (*Non-dominated Sorting Genetic algorithm II*) [12] est quant à elle basée sur la distance de *crowding* [27]. Cette mesure estime la densité d'une solution par rapport au volume d'hyper-rectangle défini par ses plus proches voisins directs, une valeur de diversité infinie étant donc affectée aux solutions extrêmes.

(c) Histogrammes

Ce type d'approche consiste à partitionner l'espace de recherche par une hyper-grille. La densité autour d'une solution est alors tout simplement estimée par le nombre d'éléments de la population avec laquelle elle partage la même case de la grille. Cette hyper-grille peut être statique, mais elle peut également être adaptée au cours de la recherche par rapport à la population courante. Cette approche est utilisée dans les algorithmes PAES (*Pareto Archived Evolution Strategy*) [40] et PESA (*Pareto Envelope based Selection Algorithm*) [10].

1.5.3 Élitisme

L'élitisme a pour but principal de préserver les solutions non-dominées trouvées par l'algorithme au cours de la recherche. De la façon la plus simple, ceci peut se faire lors de l'étape de remplacement de la P-META considérée, par le biais de la population principale. D'autre part, la totalité, ou un sous-ensemble des solutions non-dominées peuvent également être sauvegardées au sein d'une population secondaire : l'*archive* (de taille non-bornée, bornée ou fixe). La mise à jour du contenu de l'archive à l'aide de nouvelles solutions non-dominées trouvées par l'algorithme repose sur une relation de dominance telle que la dominance de Pareto ou d'autres types de relations. Parmi ces algorithmes, on trouvera le nouvel algorithm évolutionnaire proposé par Liefoghe et al. en 2009, SEEA (*Simple Elitist Evolutionary Algorithm*) [52].

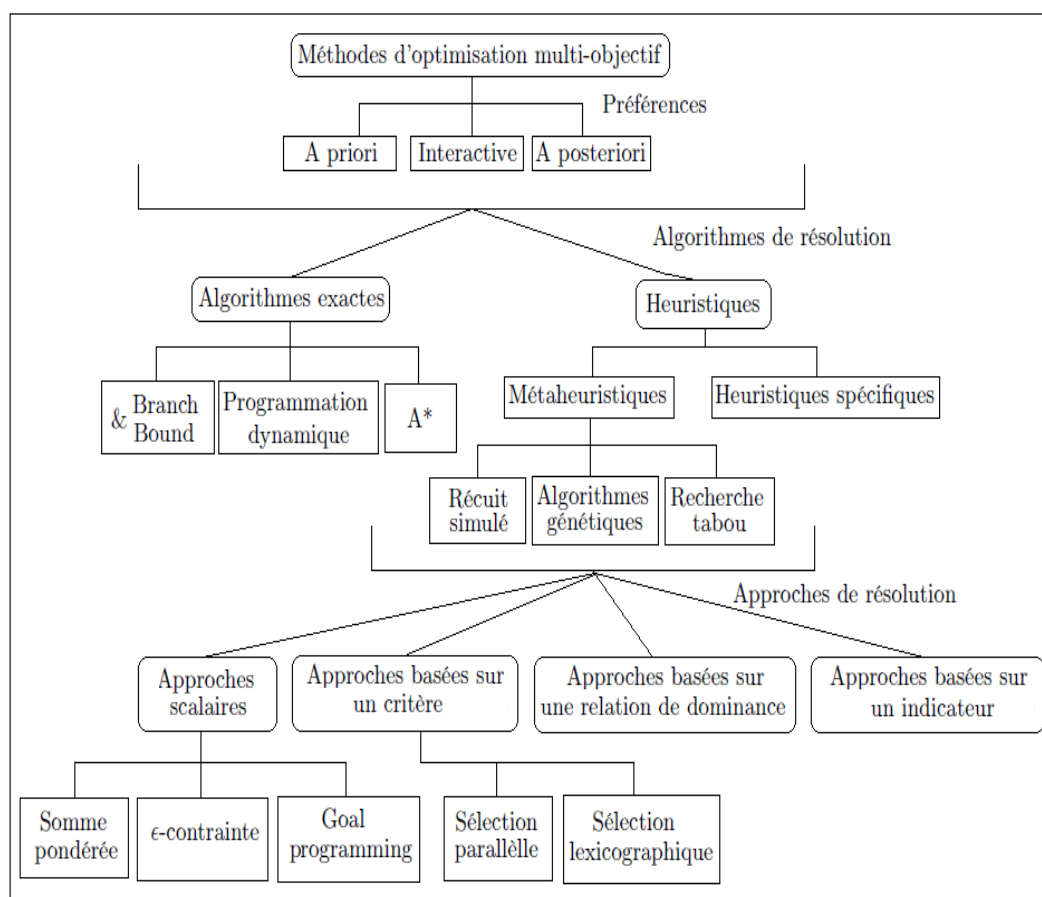


FIG. 1.12 – Classification des approches d'optimisation multi-objectif.

1.6 Recherche locale multi-objectif

Comparativement au nombre considérable d’algorithmes évolutionnaires multi-objectif, la quantité d’algorithmes de recherche locale multi-objectif est très faible, notamment en ce qui concerne les approches basées sur une relation de dominance telle que la dominance de Pareto. Dans notre travail, nous nous intéressons à ce type de métaheuristiques.

Un algorithme classique de recherche locale consiste à améliorer de façon itérative une solution arbitraire par rapport à son voisinage jusqu’à ce qu’un optimum local soit atteint (FIG. 1.13). Il s’agit donc d’une métaheuristique à base de solution unique [84]. Ainsi, la considération de ce type de méthode requiert la définition d’une structure de voisinage pour le problème traité.

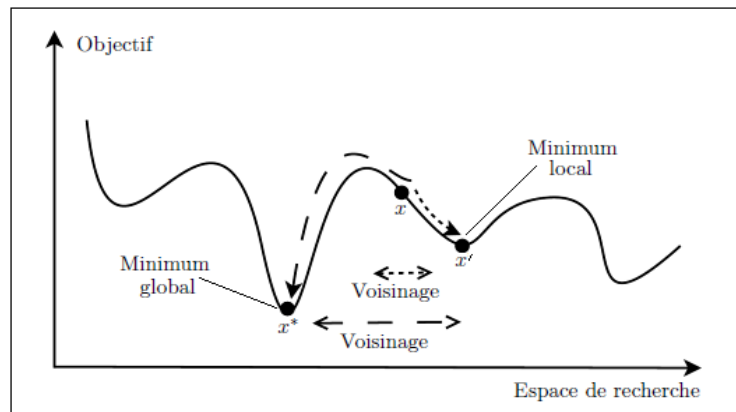


FIG. 1.13 – Recherche locale.

Définition 1.11 (Structure de voisinage) Une structure de voisinage est une fonction $N : S \rightarrow 2^S$ qui affecte un ensemble de solutions $N(x) \subset S$ à toute solution $x \in S$. Une solution x' dans le voisinage de x ($x' \in N(x)$) est un voisin de x .

Dans le cadre de l’optimisation multi-objectif, des solutions Pareto optimales doivent être trouvées. La notion d’optimum local doit être définie en termes d’optimalité Pareto.

Définition 1.12 (Solution localement Pareto optimale) Une solution $x \in S$ est une solution localement Pareto optimale par rapport à une structure de voisinage N ssi il n’existe pas de solution voisine $x' \in N(x)$ telle que $x' \prec x$.

Les approches de recherche locale proposées pour l'optimisation multi-objectif sont basées sur l'amélioration successive et indépendante d'une solution unique. De manière générale, l'approximation finale est contenue dans une archive externe où les solutions potentiellement Pareto optimales sont stockées. Ainsi, ces approches appartiennent à trois catégories différentes : *les approches scalaires*, *les approches basées sur une relation de dominance* et *les approches basées sur un indicateur de qualité*. Nous nous intéressons à la deuxième classe, qui consiste à définir le critère d'acceptation de la recherche locale sur une relation de dominance arbitraire, comme la dominance Pareto [50]. Cette classe sera notée DMLS (*Dominance-based Multiobjective Local Search*) et détaillée dans le dernier chapitre.

1.7 Problèmes d'optimisation multi-objectif

Tout comme les problèmes d'optimisation mono-objectif, les problèmes d'optimisation multi-objectif peuvent être classés en deux catégories : ceux dont les solutions représentées à l'aide de variables réelles comme les problèmes d'optimisation multi-objectif continus et ceux dont les variables sont représentées à l'aide de variables discrètes, tels que les problèmes d'optimisation multi-objectif combinatoires.

Dans les 40 dernières années, la plupart des travaux existants ont porté sur la programmation multi-objectif linéaire en variables continues. Les raisons principales de cet intérêt sont d'une part le développement de la programmation linéaire mono-objectif en recherche opérationnelle, et la facilité relative de traiter de tels problèmes polynomiaux, et d'autre part l'abondance des cas pratiques qui peuvent être formulés sous forme linéaire.

Néanmoins, l'intérêt porté aux problèmes combinatoires s'accroît depuis plusieurs années. En effet, de nombreux problèmes réels et académiques classiques ont été modélisés en problèmes d'optimisation multi-objectif combinatoires tels que : Ordonnancement [58], Voyageur de commerce [74], Routage de véhicules [66], etc. Le problème de tournées de véhicules fera l'objet du deuxième chapitre.

1.8 Conclusion

Dans ce chapitre, nous avons présenté les principales définitions nécessaires à la présentation des problèmes d'optimisation combinatoire multi-objectif ainsi que les approches de résolution de ces problèmes, notamment par des métaheuristiques.

2

Problèmes de tournées de véhicules

2.1 Introduction

Les problèmes de tournées de véhicules représentent une famille de problèmes fortement étudiée dans le cadre de la recherche opérationnelle. Ils servent à modéliser de nombreuses situations réelles, mais sont le plus souvent étudiés en ne tenant compte que d'une unique fonction objectif (le plus souvent un coût). Or, la plupart des problèmes rencontrés dans l'industrie, et notamment en logistique, sont de nature multi-objectif. L'étude et la résolution, même de manière approchée, de problèmes de tournées multi-objectif présentent de nombreux défis.

Dans ce chapitre, nous nous intéressons aux problèmes de tournées de véhicules. Une description générale de ces problèmes est présentée, plus particulièrement les problèmes multi-objectif et les principaux problèmes dérivés, ainsi que les approches de résolution de ces problèmes.

2.2 Problèmes de tournées de véhicules

Le problème de tournées de véhicules (*Vehicle Routing Problem VRP*) a été introduit en 1959 par Dantzig et Ramser [11]. Étant donné un ensemble de clients à servir et une flotte de véhicules de capacité limitée, le but est de construire un ensemble de tournées, pour un nombre fini de véhicules, visitant ces clients, commençant et finissant à un dépôt. Dans ces tournées, un client doit être desservi une seule fois par un seul véhicule et la capacité de transport d'un véhicule pour une tournée ne doit pas être dépassée.

La figure 2.1 montre un exemple de tournées de véhicules avec trois véhicules.

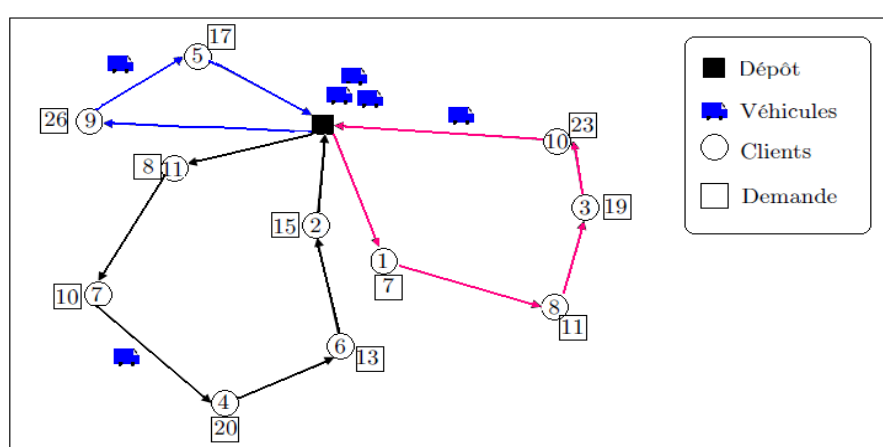


FIG. 2.1 – Vehicle Routing Problem (VRP)

On peut définir un problème de tournées de véhicules en précisant les éléments suivants et leur nature : le réseau, la demande, la flotte, le coût et l'objectif [31].

2.2.1 Réseau

Le réseau peut être symétrique ou asymétrique. Il est représenté sous la forme d'un graphe où les sommets peuvent être les villes à visiter, les clients à servir, les points de passage, les dépôts, etc. Les arcs (arêtes) représentent les liaisons existant entre ces derniers, et peuvent être des routes, des canalisations, etc. Une action de collecte ou de livraison peut être définie le long des liaisons. Aux arcs (arêtes) sont souvent associées des valeurs numériques qui indiquent les longueurs des liaisons ainsi que les temps de parcours qui peuvent dépendre du type de véhicule utilisé et de la période traversée.

2.2.2 Flotte

La première caractéristique de la flotte est le nombre de véhicules la composant. Ce nombre peut être fixe ou non. La flotte peut être homogène (composée d'un seul type de véhicules) ou hétérogène. Les véhicules composant la flotte peuvent avoir plusieurs caractéristiques qui limitent ou conditionnent leur utilisation telles que : dépôt associé, capacité...

2.2.3 Demande

La demande peut être fixe ou stochastique (incertaine). Elle peut être associée soit aux noeuds (problèmes de tournées sur les noeuds), soit aux arcs (problèmes de tournées sur les arcs). Enfin, la demande peut porter sur un seul produit ou sur plusieurs.

Le plus souvent, on parle de demande lorsque le problème traité est un problème de distribution, i.e., un problème où il s'agit de livrer à chaque noeud ou chaque arc une certaine quantité de marchandise ou de produit.

2.2.4 Coûts

Généralement, le coût est fixe pour le véhicule et variable pour son utilisation en fonction de la distance parcourue ou du temps écoulé. Le coût prend aussi en compte les pénalités de service si les clients sont servis en retard ou de manière incomplète. Un gain peut être associé aux noeuds ou aux liaisons ; le gain étant collecté lors de la visite du noeud ou de la liaison.

2.2.5 Objectif

Les plus couramment utilisés sont :

- Minimiser la longueur totale parcourue.
- Minimiser la durée totale du parcours, le temps de retard, le temps d'attente, etc.
- Minimiser le coût total du parcours.
- Minimiser la somme des coûts fixes et des coûts variables de l'utilisation des véhicules.
- Minimiser la taille de la flotte.
- Maximiser la qualité de service offert.
- Maximiser le profit collecté par la visite chez des clients choisis.

Souvent, il est question de considérer simultanément plusieurs objectifs pouvant être contradictoires. Toutefois, il est possible de les combiner dans la définition de problèmes de tournées multi-objectif.

2.3 Domaines d'application

L'application principale des problèmes de tournées de véhicules est la distribution, ou la collecte de biens depuis un ou plusieurs dépôts, et leurs distributions à des clients dispersés géographiquement. La livraison de colis, la collecte du lait, la collecte d'argent (distributeurs) sont les exemples les plus connus. Ils font apparaître une contrainte de capacité sur les véhicules utilisés lors des tournées. Le problème de tournées de véhicules apparaît aussi dans la distribution de services : visites à domicile de médecins, de représentants de commerce, d'agents de maintenance. Dans ces derniers cas, la notion de véhicule et celle de capacité peuvent être remplacées par d'autres. Par exemple, notre représentant de commerce voudra planifier ses visites hebdomadaires. Supposons qu'il travaille cinq jours par semaine, qu'il rentre chez lui tous les soirs et que la durée et/ou l'heure de ses visites est fixée ou estimée. Il doit alors résoudre l'équivalent d'un problème de tournées à cinq véhicules avec des contraintes annexes de temps.

Le problème de ramassage de déchets et la planification de tournées de cars scolaires en milieu urbain sont d'autres domaines d'application des problèmes de tournées de véhicules. Dans le premier problème, des déchets doivent être collectés dans les rues donc il s'agit d'un problème de tournées sur les arcs. Quant au deuxième problème, il s'agit d'un groupe d'élèves répartis dans l'espace qui doit recevoir un service de transport public depuis le lieu de résidence jusqu'à l'école. Le problème est de trouver une série de tournées de cars scolaires qui assure que le service est équitable pour tous les élèves admissibles.

2.4 VRP déterministe

Le problème statique (ou déterministe) de tournées de véhicules consiste à déterminer en minimisant le coût, un ensemble de tournées pour un nombre limité de véhicules, commençant et finissant à un dépôt, de telle façon que chaque client soit visité exactement une fois par un véhicule, et que la somme des demandes provenant des clients sur une tournée

ne dépasse pas la capacité du véhicule qui les dessert. Dans ce cas, toute l'information (données du problème) concernant la planification des routes est supposée connue avant que le processus de planification commence. Cette information ne change pas après que les routes aient été construites. Cette information inclut tous les attributs des clients tels que la position géographique, le temps de service passé sur place, et la demande de chaque client (quantité à collecter ou à livrer). En plus, les informations concernant les temps de trajets nécessaires pour relier les clients à servir doivent être connues ou calculables par le planificateur.

Formulation mathématique

Il existe de nombreuses formulations du problème de tournées de véhicules. La formulation suivante est tirée de Rego et Roucairol [73].

Mathématiquement, le VRP statique se définit sur un graphe $G = (N, A)$ où $N = \{0, \dots, n\}$ est un ensemble de sommets avec le sommet 0 fixé comme dépôt et $N' = N \setminus \{0\}$ est l'ensemble des clients à satisfaire. $A = \{(i, j) | i, j \in N \text{ et } i \neq j\}$ est l'ensemble des arcs, à chaque arc est associé un coût non négatif c_{ij} qui peut représenter la distance du voyage ou la durée du voyage entre i et j . Une flotte de m véhicules est disponible, ayant la même capacité de transport Q (véhicules homogènes).

Les contraintes prises en compte dans cette formulation :

- une contrainte de capacité : à chaque sommet $i \in N'$ est associé un poids d_i non négatif représentant la demande, la somme des poids d'une tournée ne doit pas dépasser la capacité Q du véhicule ;
- une contrainte de temps total : le temps total d'une tournée (somme des temps de voyage entre les sommets et des temps d'arrêt à ces derniers) ne doit pas dépasser une borne T .

Données.

n : nombre de clients ;

m : nombre de véhicules ;

Q : capacité d'un véhicule ;

T_k : temps maximal de la tournée du véhicule k ;

d_i : demande du client i , ($d_0 = 0$) ;

t_i^k : temps nécessaire au véhicule k pour charger ou décharger au client i ;

t_{ij}^k : temps nécessaire au véhicule k pour voyager du client i au client j ;

c_{ij} : coût ou distance du voyage du client i au client j .

Variables de décision.

$$x_{ij}^k = \begin{cases} 1 & \text{si le véhicule } k \text{ voyage directement du client } i \text{ au client } j \\ 0 & \text{sinon} \end{cases}$$

Formulation.

$$\text{Minimiser } \sum_{i=1}^n \sum_{j=1}^n c_{ij} \sum_{k=1}^m x_{ij}^k \quad (2.1)$$

Sous

$$\sum_{i=0}^n \sum_{k=1}^m x_{ij}^k = 1 \quad j = 1, \dots, n \quad (2.2)$$

$$\sum_{j=0}^n \sum_{k=1}^m x_{ij}^k = 1 \quad i = 1, \dots, n \quad (2.3)$$

$$\sum_{i=0}^n x_{ip} - \sum_{j=0}^n x_{pj} = 0 \quad k = 1, \dots, m; \quad p = 0, \dots, n \quad (2.4)$$

$$\sum_{i=0}^n d_i \left(\sum_{j=1}^n x_{ij}^k \right) \leq Q \quad k = 1, \dots, m \quad (2.5)$$

$$\sum_{i=0}^n t_i^k \sum_{j=0}^n x_{ij}^k + \sum_{i=0}^n \sum_{j=0}^n t_{ij}^k x_{ij}^k \leq T_k \quad k = 1, \dots, m \quad (2.6)$$

$$\sum_{j=1}^n x_{0j}^k \leq 1 \quad k = 1, \dots, m, \quad (2.7)$$

$$\sum_{i=1}^n x_{i0}^k \leq 1 \quad k = 1, \dots, m, \quad (2.8)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij}^k \leq |S| - 1 \quad \text{pour tout } S \subset N, \quad 2 \leq |S| \leq n - 1 \quad (2.9)$$

$$x_{ij}^k \in \{0, 1\} \quad i = 0, \dots, n, \quad j = 0, \dots, n, \quad k = 1, \dots, m \quad (2.10)$$

La fonction objectif (2.1) consiste à minimiser le coût total de transport.

Les contraintes (2.2) et (2.3) assurent que chaque client n'est visité qu'une seule fois par un et un seul véhicule.

La contrainte (2.4) assure que si le véhicule atteint un noeud il doit impérativement le quitter (continuité de la tournée).

(2.5) est la contrainte de capacité du véhicule.

(2.6) est la contrainte de la durée totale d'une tournée.

Les contraintes (2.7) et (2.8) assurent le non dépassement de la disponibilité d'un véhicule, un véhicule ne sort du dépôt et n'y revient qu'une seule fois.

(2.9) assure les contraintes d'élimination de sous tours (contraintes de connexité).

2.5 Variantes du VRP

Outre, la conservation de l'intégrité du problème, le fait d'ajouter, de modifier ou de supprimer des contraintes peut nous faire passer d'un problème à un autre problème dérivé du VRP. Dans ce qui suit nous présentons les principaux problèmes dérivés du VRP.

- **CVRP** (Capacitated Vehicle Routing Problem)

Un problème CVRP consiste à affecter chaque client à une tournée effectuée par un seul véhicule de capacité finie. La demande cumulée des clients se trouvant sur la même tournée ne doit pas dépasser la capacité du véhicule utilisé.

- **VRPTW** (Vehicle Routing Problem with Time Windows)

Le VRPTW est un des problèmes les plus étudiés. Dans un VRPTW, de nouvelles contraintes temporelles sont ajoutées : chaque client doit être servi dans un intervalle de temps durant lequel il est disponible pour être visité.

- **VRPB** (Vehicle Routing Problem with backhauls)

Il comporte des clients auxquels des produits doivent être livrés et des fournisseurs dont les marchandises doivent être amenées au dépôt. Donc, pour éviter de réarranger les produits transportés dans le véhicule sur une tournée, les clients doivent être visités avant les fournisseurs.

- **VRPPD** (Vehicle Routing Problem with Pickup and Delivery)

Le problème de collecte et de livraison a les mêmes propriétés que le VRP. Il ajoute à celles-ci le fait que chaque client introduit deux positions géographiquement différentes : la première pour le ramassage du produit et la deuxième pour la livraison de ce dernier. Ceci va directement induire une contrainte de précedence à ajouter au problème

du VRP classique sachant que, dans une tournée, chaque opération de livraison doit être précédée par l'opération de ramassage respective.

- **MDVRP** (Multi-Depot Vehicle Routing Problem)

Dans ce type de problème, plusieurs dépôts géographiquement distribués existent. Une tournée est assurée par un véhicule qui part et revient au même dépôt initial.

- **VRPHF** (Vehicle Routing Problem with Heterogeneous Fleet)

La seule différence entre un VRPHF et un VRP est que la flotte de véhicule est hétérogène. Une flotte de véhicules hétérogène est composée de véhicules qui sont de types différents. Les véhicules peuvent être différenciés par leurs coûts de transport, leurs capacités de transport, leurs vitesses, leurs tailles,...

- **OVRP** (Open Vehicle Routing Problem)

La différence entre un OVRP et un VRP est que dans le premier les véhicules ne sont pas tenus de retourner au dépôt. Dans le cas où il le sont, il rebrousse chemin en revisitant les clients qui leurs sont affectés dans l'ordre inverse. C'est pourquoi les parcours des véhicules sont tous des chemins ouverts.

- **SDVRP** (Split Delivery Vehicle Routing Problem)

Dans ce genre de problème, la demande du client peut être satisfaite sur plusieurs tournées ; pour cela un client peut être visité plusieurs fois si cela est nécessaire. Contrairement aux autres, la demande d'un ou plusieurs clients peut excéder la capacité du véhicule.

- **DVRP** (Dynamic Vehicle Routing Problem)

Le DVRP illustre mieux la réalité. Dans les problèmes de transport réels dans la société d'aujourd'hui la composante dynamique est très importante, elle s'exprime surtout par les apparitions de nouveaux clients, de nouvelles demandes, ou de pannes des véhicules de transport. Dans ce genre de situation, le décideur doit réorganiser, en un temps acceptable, la planification des itinéraires de sa flotte de véhicules pour répondre aux nouvelles demandes urgentes.

- **SVRP** (Stochastic Vehicle routing Problem)

Un problème VRP est dit stochastique si au moins un de ses éléments est aléatoire, c'est-à-dire avec une certaine incertitude. Ces éléments peuvent être une ou plusieurs demandes des clients, les temps ou les coûts de transport, . . . Le problème avec les demandes stochastiques est celui le plus étudié dans la littérature. La majorité des recherches suppose que les demandes aléatoires suivent une loi de distribution normale.

2.6 Approches de résolution du VRP et ses variantes

- **CVRP**

Toth et Vigo [89] présentent une revue des algorithmes exacts utilisés pour résoudre le CVRP symétrique. Dans leur papier, seulement les restrictions sur la capacité des véhicules sont imposées et l'objectif consiste à minimiser le coût total (le nombre de route et/ou leur longueur ou le temps de route nécessaire pour servir tous les clients). Les auteurs présentent deux algorithmes de «*branch and bound*» pour résoudre ce problème.

L'article d'Achuthan et al. [1] traite lui aussi du CVRP à l'aide de la méthode génération de coupe. Plus spécifiquement, ils développent de nouvelles coupes et une procédure de recherche permettant d'identifier les violations comme par exemple, la création des sous-tours.

Étant donné la complexité NP-difficiles des problèmes de tournées de véhicules et les algorithmes exacts qui ne réussissent toujours pas à trouver des solutions pour des problèmes de grandes tailles, plusieurs chercheurs se sont penchés vers des méthodes approchées.

Baker et Ayechev [3] utilisent un algorithme génétique pour résoudre le CVRP. Après avoir trouvé des routes individuelles intéressantes, les tournées sont améliorées en utilisant la méthode 2-OPT suivie du 3-OPT.

L'heuristique de Taillard [83] consiste essentiellement à décomposer le problème en sous-problèmes. L'auteur propose deux méthodes distinctes de partitionnement. Chacun des sous-problèmes est résolu en utilisant une approche taboue. Après un certain nombre d'itérations, les sous problèmes sont réunis pour reformer le problème entier. Ils ont re-

marqué que le type d'algorithme le plus utilisé pour résoudre le problème de tournées de véhicule est celui de la recherche tabou.

Deux autres articles présentent aussi des résultats très intéressants pour le problème de VRP à un seul dépôt, il s'agit de Prins [67] et Toth et Vigo [89]. Prins [67] privilégie l'algorithme génétique tandis que Toth et Vigo [89] adoptent la recherche taboue.

Pour les problèmes de grande taille Reimann et al. [70] proposent une heuristique composite inspirée de l'algorithme d'optimisation par colonie de fourmis.

• VRPTW

Plusieurs auteurs traitent le VRPTW sous différentes facettes dans la littérature. Par exemple, Hwang [29] présente une interface graphique permettant de résoudre des problèmes VRPTW grâce à un algorithme génétique. Li et Lim [49] utilisent un algorithme de recuit simulé pour résoudre ce problème.

Berger et Barkaoui [5] développent un nouvel algorithme génétique. L'approche est basée sur l'évolution simultanée de deux populations de solutions et une relaxation partielle de quelques contraintes.

Les auteurs Lau et al. [46] traitent le m-VRPTW dans lequel un nombre limité de véhicules est disponible. Les auteurs proposent une approche taboue avec une liste permettant de conserver les clients non visités ainsi qu'un mécanisme introduisant un nouveau véhicule afin de maximiser le nombre de clients visités dans cette route tout en respectant la capacité du véhicule. De plus, ils relaxent les fenêtres de temps en permettant des retards mais avec un coût de pénalité associé à ce retard.

Lorsqu'il est possible de violer la contrainte des fenêtres de temps on parle du problème de tournées avec fenêtres de temps souples (*Vehicle Routing Problem with Soft Time Window constraints (VRPSTW)*). La contrainte de fenêtre de temps est alors relaxée permettant de livrer avant ou après le début et la fin de la fenêtre de temps. Par contre, une pénalité est introduite reflétant l'insatisfaction des clients non desservis dans leur fenêtre de temps. Ioannou et al. [30] aborde ce problème dans leur papier. Ils proposent un générateur de problèmes de tournées de véhicules où chaque problème est caractérisé par un certain nombre de clients pour lesquels on peut violer la fenêtre de temps. Chacun des problèmes est résolu avec une approche de construction soit le voisinage le plus proche.

• VRPHF

Le papier de Renaud et Boctor [71] présente le problème de tournées avec une flotte de véhicules hétérogènes. La flotte peut être composée de véhicules avec des capacités différentes. Les auteurs utilisent une heuristique de balayage pour générer différentes routes. Tarantilis et al. [88] abordent le même problème d'une autre façon. Dans leur cas, la flotte comporte plusieurs types de véhicules ayant chacun une capacité variée. Par contre, le nombre de véhicule de chaque type est limité. La flotte est donc fixe, c'est-à-dire qu'on ne considère pas la possibilité pour les entreprises de varier la composition de la flotte de véhicules. L'article décrit une méthode basée sur un seuil d'acceptation appelé «backtracking adaptive threshold accepting» (BATA).

• VRPFL

Arunapum et al. abordent une variante du VRP considérant des chargements complets, il s'agit du Vehicle Routing with Full Truckload (VRPFL). Ils utilisent une méthode exacte de séparation et évaluation progressive pour résoudre ce problème.

Gronalt et al. [26] résolvent eux aussi ce problème avec un algorithme de séparation et évaluation progressive. Par ailleurs, ils proposent une relaxation du problème pour générer des bornes inférieures utilisées pour évaluer quatre nouvelles heuristiques.

• VRPDP

Plusieurs articles récents traitent le problème de tournées de véhicules avec ramassage et livraison. Le papier de Parragh et al. [62, 63] est le premier volet d'une enquête exhaustive sur ces problèmes. Deux classes de problèmes peuvent être distinguées. La première classe, portant sur le transport de marchandises à partir du dépôt et des clients. Cette classe est notée problèmes de tournées de véhicules avec retours (VRPB). Quatre sous-types peuvent être considérés, à savoir les problèmes de tournées de véhicules avec retours en cluster (VRPCB), le problème de tournées de véhicule avec retours mixte (VRPMB), les problèmes de tournées de véhicules avec livraison divisible et ramassage (VRPDDP - les clients les plus exigeants de livraison et de ramassage peuvent être visités à deux reprises), et les problèmes de tournées de véhicules avec cueillette et livraison simultanée (VRPSDP - les clients exigent que les services doivent être visités exactement une seule

fois). La seconde classe, traitée dans la deuxième partie de cette enquête, se réfère à tous ces problèmes lorsque des marchandises sont transportées entre ramassage et livraison. Ce sont les problèmes de tournées de véhicules avec ramassage et livraison (PDVRP - un point de ramassage et des points de livraison), le problème classique de ramassage et livraison (PDP), et le problème de transport des passagers (Dial-A-Ride) (DARP). Des formulations mathématiques sont données pour tous les sept types de VRP avec cueillette et livraison, des méthodes de résolution exactes, heuristiques et métaheuristiques sont discutées.

2.7 VRP du mono-objectif au multi-objectif

Pour améliorer l'aspect réel d'un problème il faut considérer plusieurs objectifs. Dans le cas des problèmes de tournées, les objectifs peuvent être classés selon la composante du problème sur laquelle ils portent : la tournée, les noeuds ou arcs et les ressources. La figure 2.2 regroupe les différents objectifs rencontrés dans la littérature [31].

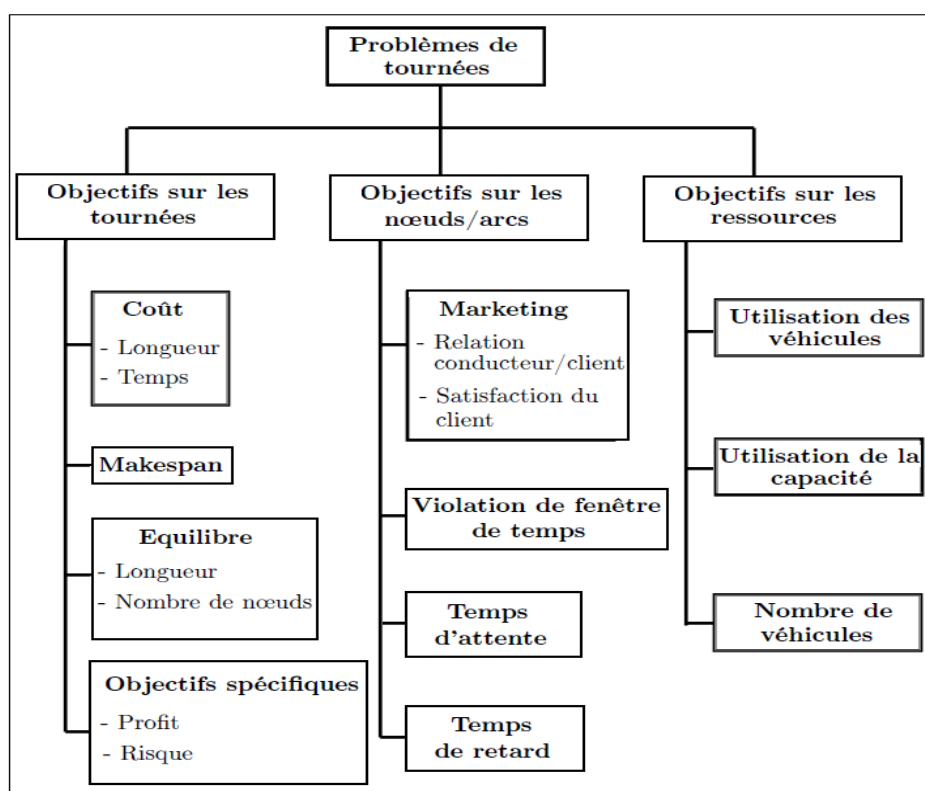


FIG. 2.2 – Différents objectifs pour le VRP

2.7.1 Objectifs sur les tournées

Coût : La plupart des études cherchent à minimiser le coût total du parcours. Ce coût peut s'exprimer en fonction de la distance parcourue, du temps nécessaire, du nombre de clients visités,...

Makespan : Il existe des études où la minimisation du coût est ignorée, elle est écartée au profit de la minimisation du makespan, i.e., la durée de la plus longue tournée.

Équilibrage : Un autre type d'objectif naturel est celui qui permet d'induire un équilibrage entre les tournées. L'objectif peut être introduit pour apporter une notion d'équité entre les travailleurs, les clients, ... Pour prendre en compte l'équilibrage entre les tournées, il est nécessaire de définir le travail ou la charge associé(e) à une tournée. Le travail d'une tournée peut s'exprimer en fonction du nombre de clients visités, de la quantité de marchandise livrée, de la longueur de la tournée, du temps nécessaire,...

Objectifs spécifiques : Certains objectifs peuvent être définis en fonction de la nature du problème ou du contexte où s'inscrit le problème (par exemple, pour un problème où un gain est associé aux différents noeuds du graphe). Ce gain peut être collecté en visitant le noeud, mais il n'est pas nécessaire de visiter tous les noeuds. Dans ce modèle, la maximisation du gain est prise en compte. Dans le cas du transport de produits dangereux, un risque est associé au parcours qu'emprunte un véhicule. Du fait des conséquences pour les habitants et l'environnement des régions traversées qui seraient catastrophiques en cas d'accident au vu des produits transportés, il est nécessaire de minimiser le risque d'accident le long des tournées.

2.7.2 Objectifs sur les noeuds et les arcs

La plupart des études qui incorporent des objectifs sur les noeuds ou les arcs traitent de problèmes qui disposent de fenêtres horaires. Dans ce cas, soit les fenêtres de temps sont ignorées et remplacées par un objectif qui minimise le nombre de fenêtres de temps violées, soit le temps d'attente des clients et/ou du conducteur exprimé par les avances et retards par rapport aux bornes de la fenêtre horaire est comptabilisé, ou ces deux objectifs sont pris en compte en même temps.

2.7.3 Objectifs sur les ressources

Les ressources rencontrées dans la littérature sont les véhicules et les marchandises. Un objectif couramment étudié est la minimisation du nombre de véhicules. Cet objectif peut avoir une signification économique : moins il y a de véhicules, moins il y a d'investissement (achats, essence, salaires des conducteurs,...). D'autres objectifs sur les véhicules peuvent être définis pour maximiser leur rentabilité en temps d'utilisation ou en capacité.

Les objectifs liés à la marchandise peuvent être introduits du fait de la nature de la marchandise. Par exemple, s'il s'agit d'une denrée périssable, il peut être utile d'inclure un objectif qui minimise la détérioration de la marchandise.

2.8 Approches de résolution du VRP multi-objectif

La plupart des études utilisent des méthodes scalaires pour traiter la présence de plusieurs objectifs. Une majorité utilise la méthode d'agrégation [65, 66, 82, 47, 9, 60, 97]. D'autres méthodes scalaires comme le but programmé [65, 66] et l'approche ϵ -contrainte [59] sont aussi utilisées. Dans la dernière étude, les auteurs essaient les différents nombres de véhicules possibles et pour un nombre de véhicules fixé cherchent à minimiser le makespan. Une approche par méthode lexicographique a aussi été proposée [38, 39]. Les autres études emploient des approches Pareto.

De nombreuses études utilisent des heuristiques spécifiques. Ces méthodes sont le plus souvent couplées avec des approches de résolution scalaires et notamment des agrégations. Corberan et al. [9] utilisent une méthode de recherche par dispersion. Dans [59], la méthode d'optimisation employée est une recherche Tabou. El-Sherbeny [17] utilise le recuit simulé multi-objectif proposé par Ulungu et al. [90]. Les autres études emploient des algorithmes évolutionnaires. Dans le cas de Sessomboon et al. [77], de Lacomme et al. [44] et Tan et al. [87], l'algorithme génétique est hybridé avec une méthode de descente. Paquete et al. [60] proposent une méthode de descente en deux phases. Dans un premier temps, le problème est résolu de manière mono-objectif, puis lorsqu'un optimum local est atteint, les deux objectifs du problème sont combinés à l'aide d'une agrégation et le problème ainsi obtenu est résolu par une méthode de descente. Les tableaux 2.1 et 2.2 résument les différentes études de la littérature des problèmes de tournées de véhicules multi-objectif.

Auteurs	Problème	Méthode	Tournées	Nœuds/Arcs	Ressources
El-Sherbeny[17]	Problème de tournées de véhicules adapté au cas d'une compagnie de transport belge.	Approche Pareto. Recuit simulé.	Min le temps total. Equilibrage (longueur). Max la flexibilité.	Min les temps d'attente.	Min le nombre de camions. Min le nombre de camions couverts. Min le nombre de camions découverts. Min les temps de travail inusités.
Geiger[22]	Problème de tournées de véhicules avec fenêtres horaires.	Approche Pareto. Algorithme génétique.	Min la distance totale.	Min la violation des fenêtres de temps. Min le nombre de violations.	Min le nombre de véhicules.
Rahoual et al.[68] Rahoual et Djoukhdjoukh[69]	Problème de tournées de véhicules avec fenêtres horaires.	Approche Pareto. Algorithme génétique.	Min la distance totale	Min le nombre de contraintes violées.	Min le nombre de véhicules.
Ribeiro et Lourenço[72]	Problème de tournées de véhicules avec périodes.	Agrégation. Recherche locale itérée.	Min la distance totale. Equilibrage (nbre de clients visités).	Marketing : relation conducteur /client.	
Corboran et al.[9]	Tournées de cars scolaires en zone rurale.	Agrégation. Recherche par dispersion.	Min le makespan.		Min le nombre de véhicules.
Jozefowicz et al. [33, 36, 37]	Problème de tournées de véhicules avec équilibrage des tournées.	Approche Pareto/ Algorithmes évolutionnaires multi-objectif. But programmé/ Recherche tabou.	Min la distance parcourue.		
Baràn et Schaefer[4]	Problème de tournées de véhicules avec fenêtres horaires.	Colonie de fourmis.	Min le temps de voyage total.	Min les temps totaux de livraison.	Min le nombre de véhicules.
Tan et al.[86]	Problème de tournées de véhicules avec fenêtres horaires.	Approche Pareto. Algorithme génétique hybride.	Min les longueurs totales.		Min le nombre de véhicules.

TAB. 2.1 – Problèmes de tournées de véhicules multi-objectif de la littérature.

Auteurs	Problème	Méthode	Tournées	Nœuds/Arcs	Ressources
Zografros et Androustopoulos[97]	Problème de tournées de véhicules avec fenêtres horaires pour le transport de produits dangereux.	Agrégation. Heuristique.	Min la longueur. Min le risque.		
Mourgaya[53]	Problème de tournées de véhicules multipériodique.	Heuristiques.	Régionalisation des clusters de visites associés aux véhicules. Equilibrage de la charge (par jour ou par véhicule).		
Chitty et Hernandez[8]	Problème de tournées de véhicules dynamique.	Colonie de fourmis.	Min le temps moyen total de transit. Min la variance totale du temps de transit.		
Jozefowicz et al. [32, 34, 35]	Problème de tournées couvrantes	Approche Pareto/ Algorithme génétique multi-objectif. Coopération : Algorithme génétique/ branch&cut.	Min la longueur.	Min la couverture.	
Murata et Itai[54, 55]	Problème de tournées de véhicules.	Approche Pareto. Algorithme génétique. Recherche locale.	Optimiser le makespan.		Min le nombre de véhicules.
Ombuki et al.[57]	Problème de tournées de véhicules avec fenêtres horaires.	Approche Pareto. Somme pondérée.	Min la longueur totale.		Min le nombre de véhicules.
Lacomme et al.[45]	Problème de tournées de véhicules avec capacité sur les arcs.	Approche Pareto. Algorithme génétique.	Min la longueur totale. Min le makespan.		
Pacheco et Marti[59]	Tournées de cars scolaires en zone rurale.	Méthode ϵ -contrainte. Recherche Tabou.	Min le makespan.		Min le nombre de véhicules.
Tan et al.[87]	Problème de tournées de véhicules avec camions et remorques.	Approche Pareto. Algorithme génétique hybride.	Min les longueurs totales.		Min le nombre de véhicules.

TAB. 2.2 – Problèmes de tournées de véhicules multi-objectif de la littérature.

2.9 VRP incertain

Contrairement au VRP classique où toutes les données du problème sont connues à l'avance, le VRP devient incertain si au moins un de ses éléments est incertain. Il peut être classé en différentes catégories :

- *VRP avec demandes incertaines* : les demandes des clients sont incertaines ;
- *VRP avec clients incertains* : l'existence de chaque client dans une route est incertaine ;
- *VRP avec demandes et clients incertains* : les demandes et l'existence de chaque client sont incertaines ;
- *VRP avec temps de voyage incertain* : le temps de voyage entre deux clients est incertain à cause des conditions de la route, du temps, du trafic,...

Dans ce travail nous nous intéressons au problème de tournées de véhicules avec demandes incertaines. Dans ce cas, la demande du client n'est connue de façon certaine qu'à l'arrivée chez le client. Dans la littérature, deux stratégies ont été utilisées pour résoudre ce type de problème :

- L'*optimisation a priori* où dans ce cas le véhicule qui devient vide retourne au dépôt afin de servir le client [23] ;
- La stratégie de *prévention du restockage*, qui étudie le restockage lorsqu'un véhicule est proche du dépôt [6].

Dans notre cas, nous nous plaçons dans un cadre d'optimisation a priori.

2.10 Conclusion

Dans ce chapitre nous avons présenté le problème de tournées de véhicules en général et le problème multi-objectif en particulier. Passant de sa formulation mathématique et ses variantes à son état de l'art, nous avons énuméré les différentes méthodes de résolution qui ont été utilisées pour le résoudre. Ensuite nous avons présenté brièvement le problème de tournées avec demandes incertaines auquel nous nous intéressons dans la suite.

3

Modélisation d'une information imparfaite

3.1 Introduction

Les décideurs sont souvent confrontés à des situations où les données du problème sont imprécises ou incertaines : des données sur des événements du passé qui ne pouvaient pas être connues exactement en raison d'erreurs de mesure, des données sur des événements futurs (demandes) et simplement ne peuvent pas être connues avec certitude.

Dans ce chapitre, nous introduisons quelques concepts fondamentaux tels que l'incertitude, l'imprécision, l'incomplétude et nous présentons les principales théories permettant de représenter ces connaissances imparfaites. Par la suite, de nouvelles relations de dominance seront présentées pour la prise en compte de l'incertitude.

3.2 Concepts de base

D'après Bernadette Bouchon-Meunier [7] ; nos connaissances sont souvent imparfaites et il distingue trois types : incertitude, imprécision et incomplétude. Ces trois concepts d'im-

perfection sont définis comme ci-dessous :

3.2.1 Incertitude

L'*incertitude* concerne un doute sur la validité d'une connaissance, ou le fait de ne pas connaître ou prévoir l'état de la réalité pour déterminer la valeur de vérité d'une proposition. Par exemple : « Je crois que dans la salle il y a 100 personnes mais ce n'est pas sûr... ». Elle est liée à la théorie des probabilités. Dans la littérature, on distingue deux types différents d'incertitude qui sont importants pour la modélisation des problèmes du monde réel : l'incertitude aléatoire et l'incertitude épistémique.

Incertitude aléatoire : ce type d'incertitude est défini comme "variation intrinsèque associée au système ou à l'environnement en cours d'étude" [56]. L'incertitude aléatoire d'une quantité peut être souvent distinguée d'autres types d'incertitude par sa caractérisation comme valeur aléatoire avec une distribution connue. Cette incertitude est également connue sous la dénomination : incertitude stochastique, incertitude de type A, incertitude irréductible, variabilité, incertitude objective.

Incertitude épistémique : ce type d'incertitude ne décrit pas l'incertitude concernant les résultats d'un certain événement aléatoire à cause des variations du système, mais l'incertitude des résultats dûs à "n'importe quel manque d'information dans n'importe quelle phase du processus de modélisation" [56]. L'incertitude épistémique n'est pas une propriété intrinsèque du système. Un gain d'information sur le système ou les facteurs environnementaux peut conduire à une réduction de ce type d'incertitude. Cette incertitude est également connue sous la dénomination : incertitude subjective, incertitude de type B, incertitude réductible, ignorance.

3.2.2 Imprécision

L'*imprécision* concerne la difficulté dans l'énoncé d'une connaissance, ou fait référence à la description incomplète d'un état de la réalité par une proposition. Par exemple : « Dans la salle il y a environ une centaine de personnes », ou « Le poids de la table est d'environ 25kg », ou « Jean est grand ». Modéliser l'imprécision consiste à formaliser les termes de «environ», «centaine» ou «grand».

3.2.3 Incomplétude

L'*incomplétude* porte sur l'absence de connaissances ou sur des connaissances partielles sur certaines caractéristiques du système. Elle est due à une incomplétude dans les données, à l'absence d'une connaissance explicite ou à l'existence d'une connaissance générale. Par exemple : « Pour une instance d'une base de données la valeur de l'attribut Nom n'est pas remplie ».

3.3 Modélisation des imperfections

Les théories mathématiques et les techniques capables de modéliser, de gérer et de contrôler des données imparfaites, font l'objet de plusieurs travaux dans la littérature. Les théories employées sont nombreuses, citons : la théorie des probabilités [42], la théorie des sous-ensembles flous [92], la théorie des possibilités [93, 14] et la théorie des fonctions de croyance [13, 78].

3.3.1 Théorie des probabilités

La théorie des probabilités est l'outil le plus traditionnel qui permet de modéliser une information imparfaite. La notion de probabilité est liée à celle d'expérience aléatoire dont on ne peut prédire avec certitude le résultat. Cette expérience est un événement représentant une proposition logique relative au résultat. Cet événement est un élément de l'ensemble Ω de tous les événements possibles, appelé univers ou référentiel.

Définition 3.1 Soit $(\Omega, S(\Omega))$ un espace probabiliste associé à une expérience aléatoire, où Ω représente l'ensemble de référence contenant tous les résultats possibles et $S(\Omega)$ représente l'ensemble de toutes les parties de Ω . La probabilité de l'occurrence d'un événement A appartenant à l'ensemble $S(\Omega)$ est décrite par une mesure de probabilité $P : S(\Omega) \rightarrow [0, 1]$, vérifiant les axiomes suivants :

$$0 \leq P(A) \leq 1 \quad (3.1)$$

$$P(\Omega) = 1 \quad (3.2)$$

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i) \quad (3.3)$$

L'axiome (3.1) énonce ainsi "la probabilité que le résultat de l'expérience soit un point de A est un certain nombre compris entre 0 et 1".

L'axiome (3.2) énonce que le résultat sera un point de Ω avec une probabilité de 1.

L'axiome (3.3) énonce que, pour chaque séquence d'événements mutuellement exclusifs, la probabilité qu'au moins l'un de ces événements survienne est simplement la somme de leurs probabilités respectives.

On peut déduire de ces axiomes de base les deux propriétés suivantes :

Propriété 3.1 La connaissance de la probabilité d'un événement A détermine complètement celle de son événement contraire \bar{A} :

$$\forall A \subseteq \Omega, \quad P(A) = 1 - P(\bar{A}) \quad (3.4)$$

Propriété 3.2 En lien avec la mesure de probabilité P , et dans le cas d'un univers fini et discret formé d'un ensemble de singletons, une fonction $p : \Omega \rightarrow [0, 1]$ peut être définie par $p(\omega) = P(\{\omega\})$. Cela veut dire que la probabilité des événements peut être caractérisée aussi par la distribution de probabilité p sur un univers discret Ω .

De l'axiome de l'additivité (3.3) et en vérifiant la condition de normalisation ($\sum_{\omega \in \Omega} p(\omega) = 1$), on déduit que :

$$\forall A \subseteq \Omega, \quad P(A) = \sum_{\omega \in A} p(\omega) \quad (3.5)$$

Dans le cas où Ω est continu $P(A) = \int_A f(\omega) d\omega$ et $\int_{\Omega} f(\omega) d\omega = 1$. La fonction f est appelée densité de probabilité.

On peut distinguer deux interprétations concernant la mesure de probabilité P . La première, appelée fréquentiste, considère une probabilité $P(A)$ comme la limite d'une fréquence de l'occurrence de l'événement A lorsque l'expérience est répétée un grand nombre de fois. Elle est à la base de l'utilisation des informations statistiques. La deuxième interprétation, dite subjectiviste, exclut la notion de répétition d'une expérience aléatoire et suppose qu'on peut attacher à la probabilité $P(A)$ une valeur subjective quantifiant la croyance qu'on accorde, à l'occurrence de A , de façon à ce que P obéisse aux axiomes ci-dessus.

Limitations de la théorie des probabilités

Lorsque certains indices observés permettent d'augmenter la confiance à une hypothèse, l'axiome d'additivité (3.3) nécessite de diminuer d'autant la confiance correspondant à l'hypothèse contraire. Néanmoins, dans certains cas, des informations peuvent très bien favoriser une hypothèse sans pour autant discréditer l'hypothèse contraire. La théorie des probabilités semble donc peu adaptée à des situations où la connaissance d'un événement comme la connaissance de son contraire sont très limitées.

Exemple 3.1 Prenons comme exemple un problème de classification des objets dans une des deux classes C_1 et C_2 . Si un objet ω appartient fortement à C_1 , il aura une probabilité $p_\omega(\{C_1\})$ d'appartenance à cette classe très forte, tandis qu'il aura une valeur faible d'appartenance à C_2 . Cette constatation se justifie par la condition de normalisation : $p_\omega(\{C_1\}) + p_\omega(\{C_2\}) = 1$.

Supposons maintenant que la connaissance que l'on a des classes est incomplète et qu'il y a une troisième classe inconnue. Si l'objet ω n'appartient ni à C_1 ni à C_2 , à cause de la condition de normalisation cet objet aura certainement une probabilité d'appartenance forte à une des deux classes C_1 ou C_2 . La théorie des probabilités est donc inadaptée pour représenter une information incomplète. De plus, cette théorie est trop rigide pour exprimer le cas de l'ignorance totale.

Conclusion

La théorie des probabilités constitue un outil efficace pour la modélisation des incertitudes aléatoires et le cas où nous disposons d'une bonne connaissance des événements et de leurs événements contraires. Elle est limitée, parce qu'elle ne permet pas une modélisation souple de l'incomplétude et de l'imprécision. Nous allons introduire dans ce qui suit la notion de sous-ensemble flou qui permet de traiter l'aspect imprécis et vague des connaissances imparfaites.

3.3.2 Théorie des sous-ensembles flous (Fuzzy set)

Les sous-ensembles flous ont été introduits en 1965 par Zadeh [92] comme généralisation des ensembles classiques aux ensembles dont les frontières sont mal définies afin de résoudre le

problème posé par les connaissances exprimées symboliquement, présentant des imprécisions ou ayant un caractère vague telles que grand, moyen ou petit, et les connaissances décrites numériquement. Cette théorie repose sur l'idée d'appartenance partielle à une classe, de gradualité dans le passage d'une limite à une autre, ainsi que sur le fait qu'une classe donnée n'a pas de bornes strictes.

Définition 3.2 *Dans la théorie classique, un sous-ensemble A d'un référentiel Ω est défini par une collection d'objets possédant une ou plusieurs propriétés communes caractéristiques de ce sous-ensemble. Chaque objet ω de Ω est ainsi caractérisé par son appartenance ou non aux sous-ensembles A . Cela s'exprime par la fonction caractéristique $\mu_A : \Omega \rightarrow \{0, 1\}$:*

$$\mu_A(\omega) = \begin{cases} 1 & \text{si } \omega \in A \\ 0 & \text{si } \omega \notin A \end{cases} \quad (3.6)$$

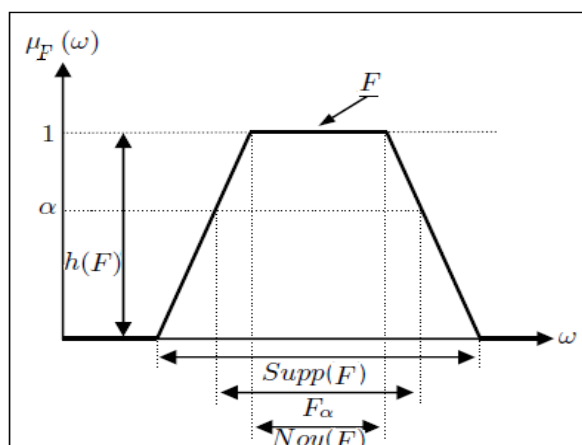
Définition 3.3 *Un sous-ensemble flou F d'un univers Ω est défini par une fonction d'appartenance $\mu_F : \Omega \rightarrow [0, 1]$ qui associe à chaque élément ω de Ω un coefficient $\mu_F(\omega)$ indiquant son degré d'appartenance à F . Plus ω s'approche de la caractérisation typique du sous-ensemble flou F , plus sa valeur d'appartenance $\mu_F(\omega)$ tend vers 1. Le sous-ensemble flou F est défini par la connaissance des couples $(\omega, \mu_F(\omega))$.*

Caractéristiques d'un sous ensemble flou :

Un sous-ensemble flou F est caractérisé par :

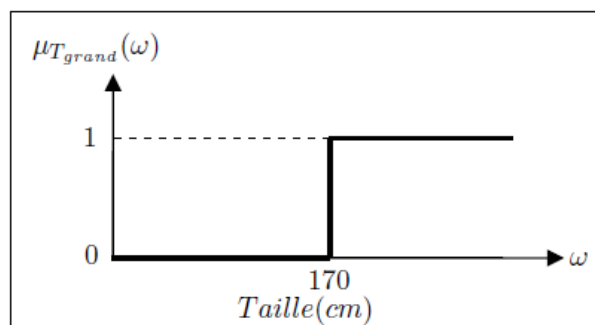
- Le *noyau* défini comme l'ensemble classique (ou net) des éléments ω appartenant totalement à F (c'est-à-dire pour lesquels $\mu_F(\omega) = 1$);
- Le *support* défini comme l'ensemble net des éléments ayant un degré d'appartenance non nul;
- La *hauteur* $h(F)$ comme la valeur $\sup_{\omega \in F} \mu_F(\omega)$. Si elle est égale à 1 on dit que F est normalisé;
- l' α -*coupe* de F (ou la coupe de niveau α), F_α , est l'ensemble net des éléments ayant une appartenance $\mu_F(\omega) \geq \alpha$.

La figure 3.1 montre ces éléments.

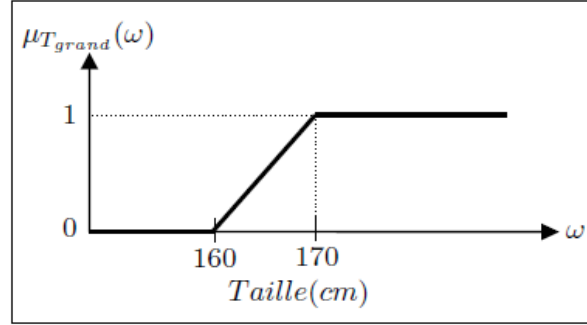
FIG. 3.1 – Quelques éléments caractéristiques de A

Exemple 3.2 Soit Ω l'ensemble des tailles des personnes vivant en Algérie. Le sous-ensemble classique grand T_{grand} correspondant aux personnes de grandes tailles, est défini par (FIG. 3.2) :

$$\mu_{T_{grand}}(\omega) = \begin{cases} 1 & \text{si } \omega \geq 170\text{cm} \\ 0 & \text{sinon} \end{cases} \quad (3.7)$$

FIG. 3.2 – Fonction d'appartenance exclusive T_{grand}

La fonction $\mu_{T_{grand}}(\omega) : \Omega \rightarrow \{0, 1\}$ traduit une appartenance tout ou rien de la taille au sous-ensemble T_{grand} . Afin d'éviter la transition brusque entre le 1 et le 0, l'idée d'appartenance graduelle de ω au sous-ensemble T_{grand} est exprimée par la théorie des sous ensembles flous. La figure 3.3 montre la fonction d'appartenance pour l'exemple de la taille.

FIG. 3.3 – Fonction d'appartenance floue T_{grand} **Propriétés des sous ensembles flous :**

Soit A et B deux ensembles flous de Ω .

1. Egalité : $A = B$ si $\forall \omega \in \Omega : \mu_A(\omega) = \mu_B(\omega)$.
2. Inclusion : $A \subset B$ si $\forall \omega \in \Omega : \mu_A(\omega) \leq \mu_B(\omega)$.
3. Intersection : $A \cap B = C$ tel que : $\forall \omega \in \Omega : \mu_C(\omega) = \min(\mu_A(\omega), \mu_B(\omega))$.
4. Union : $A \cup B = D$ tel que : $\forall \omega \in \Omega : \mu_D(\omega) = \max(\mu_A(\omega), \mu_B(\omega))$.
5. Complément d'un sous-ensemble flou : $\forall \omega \in \Omega : \mu_{A^c}(\omega) = 1 - \mu_A(\omega)$.

Définition 3.4 Un nombre flou a est un sous-ensemble flou A , convexe et normalisé, de l'ensemble des nombres réels : $\Omega = \mathbb{R}$.

Représentation $L - R$ d'un nombre flou :

Un nombre flou a est de représentation $L - R$ si et seulement si :

$$\mu_a(\omega) = \begin{cases} L\left(\frac{a-\omega}{\alpha^L}\right) & \text{si } x \leq a, \text{ avec } \alpha^L > 0 \\ R\left(\frac{\omega-a}{\alpha^R}\right) & \text{si } x \geq a, \text{ avec } \alpha^R > 0 \end{cases}$$

où L et R sont des fonctions dites de référence du nombre flou vérifiant les propriétés :

- L et R fonctions non croissantes sur $[0, +\infty[$.
- L et R fonctions symétriques : $L(\omega) = L(-\omega)$; $R(\omega) = R(-\omega)$.
- $L(0) = R(0) = 1$.

Si l'on ajoute à cette définition que $L(1) = R(1) = 0$, le support du nombre flou est fini.

Représentation plat de type $L - R$ d'un nombre flou

Un nombre flou plat de type $L - R$ est tel qu'il existe $a^L, a^R \in \mathbb{R}$, avec $a^L < a^R$ de sorte que $\mu_a(\omega) = 1 \forall \omega \in [a^L, a^R]$ et sa représentation de type $L - R$ est comme suit :

$$\mu_a(\omega) = \begin{cases} L(\frac{a-\omega}{\alpha^L}) & \text{si } \omega < a^L, \text{ avec } \alpha^L > 0 \\ 1 & \text{si } a^L \leq \omega \leq a^R, \\ R(\frac{\omega-a}{\alpha^R}) & \text{si } \omega > a^R, \text{ avec } \alpha^R > 0 \end{cases}$$

Un tel nombre flou a est désigné par $(a^L, a^R, \alpha^L, \alpha^R)_{LR}$.

Conclusion

Dans le cadre de la théorie des probabilités, l'incertitude est modélisée par des fonctions de probabilité discrètes et continues, par contre la théorie des sous-ensembles flous permet de traiter l'imprécision en considérant les paramètres aléatoires comme des ensembles flous. Cette dernière ne permet pas de modéliser l'incertitude. Par conséquent, la théorie des possibilités a été introduite pour gérer ce type d'imperfection.

3.3.3 Théorie des possibilités

La théorie des possibilités introduite par Zadeh [93], puis développée par Dubois et Prade [14], est étroitement liée à la théorie des sous-ensembles flous [92]. Elle permet de modéliser les incertitudes sur des événements, sans avoir besoin de connaissances a priori. Elle fournit un moyen de dire dans quelle mesure un événement est possible et dans quelle mesure on en est certain, sans toutefois avoir à sa disposition l'évaluation de la probabilité de cette réalisation. Ces deux affirmations sont formalisées grâce à deux mesures : la mesure de *possibilité* et la mesure de *nécessité*.

Mesure de possibilité

Définition 3.5 On peut définir la mesure de possibilité Π d'un événement A comme une fonction $\Pi : S(\Omega) \rightarrow [0, 1]$ qui associe à A un coefficient (degré de possibilité) compris entre 0 et 1 évaluant à quel point l'événement A est possible. De plus, cette fonction doit vérifier les axiomes suivants :

$$\forall A \subseteq \Omega, \quad (\Pi(\emptyset) = 0) \leq \Pi(A) \leq (\Pi(\Omega) = 1) \quad (3.8)$$

$$\forall A, B \subset \Omega, \quad \Pi(A \cup B) = \max(\Pi(A), \Pi(B)) \quad (3.9)$$

Dans l'axiome (3.8), $\Pi(A)$ quantifie dans quelle mesure l'événement A est possible : c'est-à-dire que A est tout à fait possible si la mesure de sa possibilité est égale à 1, et impossible si celle-ci est nulle.

L'axiome (3.9) exprime que la réalisation de l'un des deux événements A ou B , pris indifféremment, est affectée du même coefficient de possibilité que la réalisation de l'événement le plus possible. On peut déduire des axiomes (3.8) et (3.9) que :

$$\forall A \subset \Omega, \quad \max(\Pi(A), \Pi(\bar{A})) = 1 \quad (3.10)$$

Un événement possible A n'interdit donc pas pour autant l'événement contraire \bar{A} de l'être également, ce qui peut conduire à une situation d'ignorance totale :

$$\forall A \subset \Omega, \quad \Pi(A) = 1 \quad (3.11)$$

Mesure de nécessité

Une mesure de possibilité Π donne une information sur l'occurrence d'un événement A relatif à l'ensemble de référence Ω , mais elle ne suffit pas pour décrire l'incertitude existant sur cet événement. Par exemple, si $\Pi(A) = 1$, il est tout à fait possible que A soit réalisé mais on peut avoir en même temps :

- $\Pi(\bar{A}) = 1$, qui exprime une indétermination complète sur la réalisation de A ;
- $\Pi(\bar{A}) = 0$, qui met en évidence le fait que A seul peut être réalisé.

Pour compléter l'information sur A , une mesure de nécessité $N(A)$, duale de la mesure de possibilité, a été introduite. Elle indique le degré avec lequel la réalisation d'un événement A est certaine. On dit qu'un événement A est nécessaire si son événement contraire est impossible $\Pi(\bar{A}) = 0$, d'où le lien entre les deux mesures :

$$\forall A \subset \Omega, \quad N(A) = 1 - \Pi(\bar{A}) \quad (3.12)$$

Définition 3.6 *La mesure de nécessité est une fonction $N : S(\Omega) \rightarrow [0, 1]$ qui vérifie les axiomes suivants :*

$$\forall A \subseteq \Omega, \quad (N(\emptyset) = 0) \leq N(A) \leq (N(\Omega) = 1) \quad (3.13)$$

$$\forall A, B \subset \Omega, \quad N(A \cap B) = \min(N(A), N(B)) \quad (3.14)$$

Distribution de possibilité

Définition 3.7 *De façon similaire à la mesure de probabilité P , une mesure de possibilité Π peut être définie à partir d'une distribution de possibilité π sur les singletons de Ω dans $[0, 1]$:*

$$\Pi(\{\omega\}) = \pi(\omega), \quad \Pi(A) = \sup_{\omega \in A} \pi(\omega) \text{ avec } \sup_{\omega \in \Omega} \pi(\omega) = 1 \quad (3.15)$$

(3.15) traduit le fait que la possibilité d'un événement formé par une collection d'éléments est égale au plus grand degré de possibilité parmi ses éléments.

La mesure de nécessité peut également s'exprimer en fonction de la distribution de possibilité π associée à Π :

$$\forall A \subset \Omega, \quad N(A) = \inf_{\omega \notin A} (1 - \pi(\omega)) \quad (3.16)$$

Conclusion

Le modèle possibiliste permet de modéliser l'imprécision incluse dans une proposition A par un couple de valeurs $(N(A), \Pi(A))$. Une différence majeure entre la théorie des probabilités et la théorie des possibilités concerne un événement et son complément. La probabilité d'un événement détermine complètement son complémentaire, alors que la possibilité et la nécessité d'un événement et son complémentaire sont faiblement liées.

3.3.4 Théorie des fonctions de croyance

La théorie des fonctions de croyance, connue également sous le nom de théorie de l'évidence ou théorie de Dempster-Shafer a été introduite par Shafer en 1976 [78] à la suite des travaux de Dempster [13] sur les probabilités inférieure et supérieure. La théorie des fonctions de croyance permet de modéliser les connaissances en utilisant deux mesures floues, une mesure de crédibilité et une mesure de plausibilité. Ces mesures sont appelées fonctions de croyance et elles se placent dans un cadre plus général que la théorie des possibilités et la théorie des probabilités.

Représentation explicite des connaissances

Dans le cadre de la théorie des fonctions de croyance, on considère un univers de référence appelé le cadre de discernement Ω , composé d'un ensemble de N hypothèses supposées répondre à un problème donné :

$$\Omega = \{H_1, H_2, \dots, H_N\}$$

À partir du cadre de discernement, un référentiel de définition, noté 2^Ω , contenant tous les sous-ensembles de Ω , est défini :

$$2^\Omega = \{\{H_1\}, \{H_2\}, \{H_1, H_2\} \dots \{H_1, \dots, H_{N-1}\}, \Omega\}$$

Où le sous ensemble $\{H_i, H_j\}$ représente la proposition que la solution au problème est une des deux hypothèses, H_i ou H_j .

Un point clé de la théorie des fonctions de croyance réside dans la possibilité de représenter de manière explicite la croyance que nous avons en une hypothèse à travers les fonctions de croyance.

Définition 3.8 Une fonction de croyance peut être définie par une fonction de masse $m : 2^\Omega \rightarrow [0, 1]$, vérifiant :

$$\begin{aligned} m(\emptyset) &= 0 \\ \sum_{A \in 2^\Omega} m(A) &= 1 \end{aligned} \quad (3.17)$$

Dans l'équation (3.17), $m(A)$ s'appelle *masse de croyance* et représente le degré de croyance avec lequel une source donnée croit en la proposition A . Le degré de croyance représente le degré de précision d'une proposition par rapport à l'ensemble des connaissances d'un expert, à l'opposé du degré de vérité qui fait référence au degré de précision d'une proposition vis-à-vis de la réalité.

La masse de croyance se différencie de la notion de probabilité par le fait que la totalité de la masse de croyance est répartie non seulement sur les hypothèses singletons, H_1, H_2, H_3 , mais aussi sur les hypothèses combinées, par exemple $\{H_1, H_2\}, \{H_1, H_3\}, \{H_1, H_2, H_3\}$. Par conséquent, la masse de croyance présente une grande analogie avec la notion de distribution de probabilité, à la différence près que l'on répartit une masse unité parmi les éléments de 2^Ω .

Définition 3.9 Un élément $A \subseteq \Omega$ avec une masse $m(A) > 0$ est appelé *élément focal*.

Tous les sous-ensembles de la proposition pour lesquels la masse de croyance est non nulle sont appelés *éléments focaux*. Lorsque les éléments focaux se réduisent aux seuls singletons H_i , la notion de masse élémentaire est assimilable à celle de probabilité. La structure de croyance devient une distribution de probabilité, si elle ne contient que des singletons.

Fonctions de crédibilité et plausibilité

Une masse m peut être représentée par deux mesures non additives : une mesure de crédibilité Bel et une mesure de plausibilité Pl .

Définition 3.10 La fonction de croyance (belief function), $Bel(A)$, mesure la force avec laquelle on croit en la véracité du sous ensemble A et elle est définie comme étant la somme de toutes les masses de croyance des éléments focaux $B \in 2^\Omega$ inclus dans A (FIG. 3.4).

$$Bel : 2^\Omega \rightarrow [0, 1]$$

$$\forall A \subseteq \Omega, \quad Bel(A) = \sum_{B \subseteq A} m(B) \quad (3.18)$$

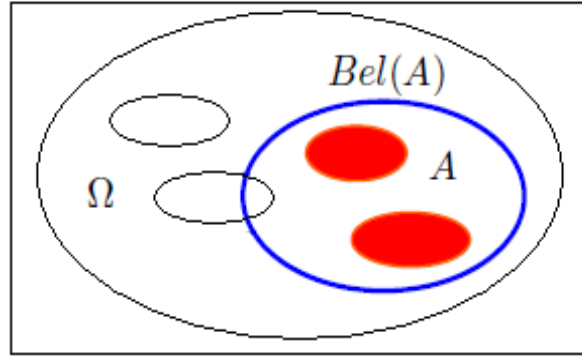


FIG. 3.4 – Fonction de croyance.

Définition 3.11 La fonction de plausibilité (plausibility function) mesure l'intensité avec laquelle on ne doute pas du sous-ensemble A , c'est-à-dire qu'elle est la somme des masses de croyance des éléments focaux dont l'intersection avec A n'est pas nulle (FIG. 3.5). Elle est définie par l'équation suivante :

$$Pl : 2^\Omega \rightarrow [0, 1]$$

$$\forall A \subseteq \Omega, \quad Pl(A) = \sum_{B \cap A \neq \emptyset} m(B) \quad (3.19)$$

La fonction de plausibilité, $Pl(A)$, est une fonction duale de la fonction de crédibilité pouvant être également définie à partir de la fonction de crédibilité de l'événement contraire de A (FIG. 3.6) :

$$Bel(A) = 1 - Pl(\bar{A}) \quad (3.20)$$

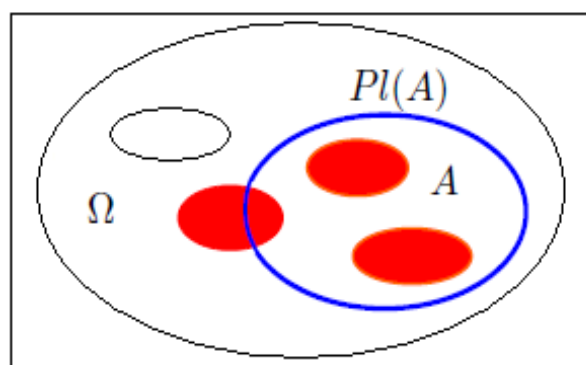
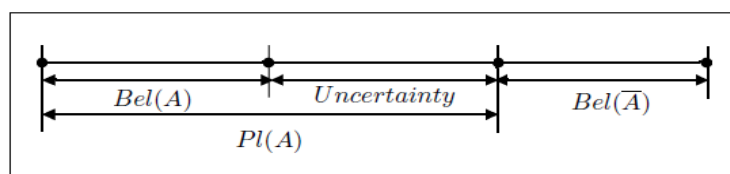


FIG. 3.5 – Fonction de plausibilité.

L'équation ci-dessus interprète le fait que, plus on augmente la croyance dans une hypothèse, moins l'hypothèse contraire devient plausible. La fonction de croyance est une mesure pessimiste, alors que la fonction de plausibilité est une mesure optimiste.

FIG. 3.6 – Croyance (Bel) et Plausibilité (Pl)

Dans le cadre de la théorie des fonctions de croyance, la croyance et la plausibilité en une hypothèse peuvent être interprétées respectivement comme les incertitudes minimum et maximum sur l'hypothèse. Ainsi, l'incertitude est représentée par l'ensemble des valeurs de l'intervalle $[Bel, Pl]$ appelé intervalle de croyance et sa longueur représente l'imprécision. Par conséquent, $Bel(A)$ et $Pl(A)$ peuvent être interprétées comme des bornes de la probabilité $P(X \in A)$.

$$Bel(A) \leq P(X \in A) \leq Pl(A)$$

Grâce à ses nombreux avantages, la théorie des fonctions de croyance a montré son utilité dans de nombreux domaines d'application tels que l'analyse de données, le diagnostic, l'aide à la décision, le traitement d'images, ...

3.4 Comparaison entre les différentes théories

Plusieurs travaux consacrés à comparer les théories mathématiques de modélisation des imperfections ont été réalisés dans la littérature. Il semblerait qu'elles ne soient pas contradictoires, et le choix de l'utilisation dépend de l'application visée.

La théorie des sous-ensembles flous permet de représenter des connaissances exprimées en langage naturel (« grand », « petit », « environ », etc.) présentant des imperfections et de réaliser une interface entre les connaissances symboliques et les descriptions numériques. Cependant, elle ne permet pas de modéliser les incertitudes. Or, comme d'ailleurs Bouchon-Meunier[7] l'affirme, l'imprécision et l'incertitude sont fortement liées.

Nous pouvons dire que la théorie des possibilités est utilisée lorsque les connaissances concernant la réalisation des événements sont insuffisantes. Elle permet de modéliser l'incertitude, l'imprécision et l'incomplétude. Cependant la représentation explicite de l'incomplétude est moins fine que dans le cadre de la théorie des fonctions de croyance. En effet, la nécessité est nulle et la possibilité est égale à 1.

Bouchon-Meunier[7] montre que la théorie des fonctions de croyance n'est pas incompatible avec les autres théories. La théorie des probabilités et la théorie des possibilités sont des cas particuliers de la théorie des fonctions de croyance, lorsque les éléments focaux sont respectivement disjoints et élémentaires (théorie des probabilités) ou emboîtés (théorie des possibilités).

Autrement dit, la théorie de l'évidence accepte toutes les répartitions possibles de la masse initiale de croyance entre les divers événements. Deux situations sont particulièrement remarquables, et conduisent à la mise en évidence des théories des possibilités et des probabilités comme cas particuliers de la théorie de l'évidence.

Dans la première, les événements ne sont affectés de masses de croyance non nulles que s'ils sont concordants, les éléments focaux sont alors emboîtés. La fonction de plausibilité a les propriétés d'une mesure de possibilité, et la fonction de croyance celles d'une mesure de nécessité.

Dans la deuxième, les croyances émises concernent des éléments de Ω pris individuellement, les éléments focaux sont donc des singletons de Ω . La fonction de plausibilité et la fonction de croyance sont identiques et ont les propriétés d'une mesure de probabilité.

La figure 3.7 représente une tentative de classification des théories de modélisation des imperfections retenus, à savoir, les probabilités, les possibilités, l'évidence.

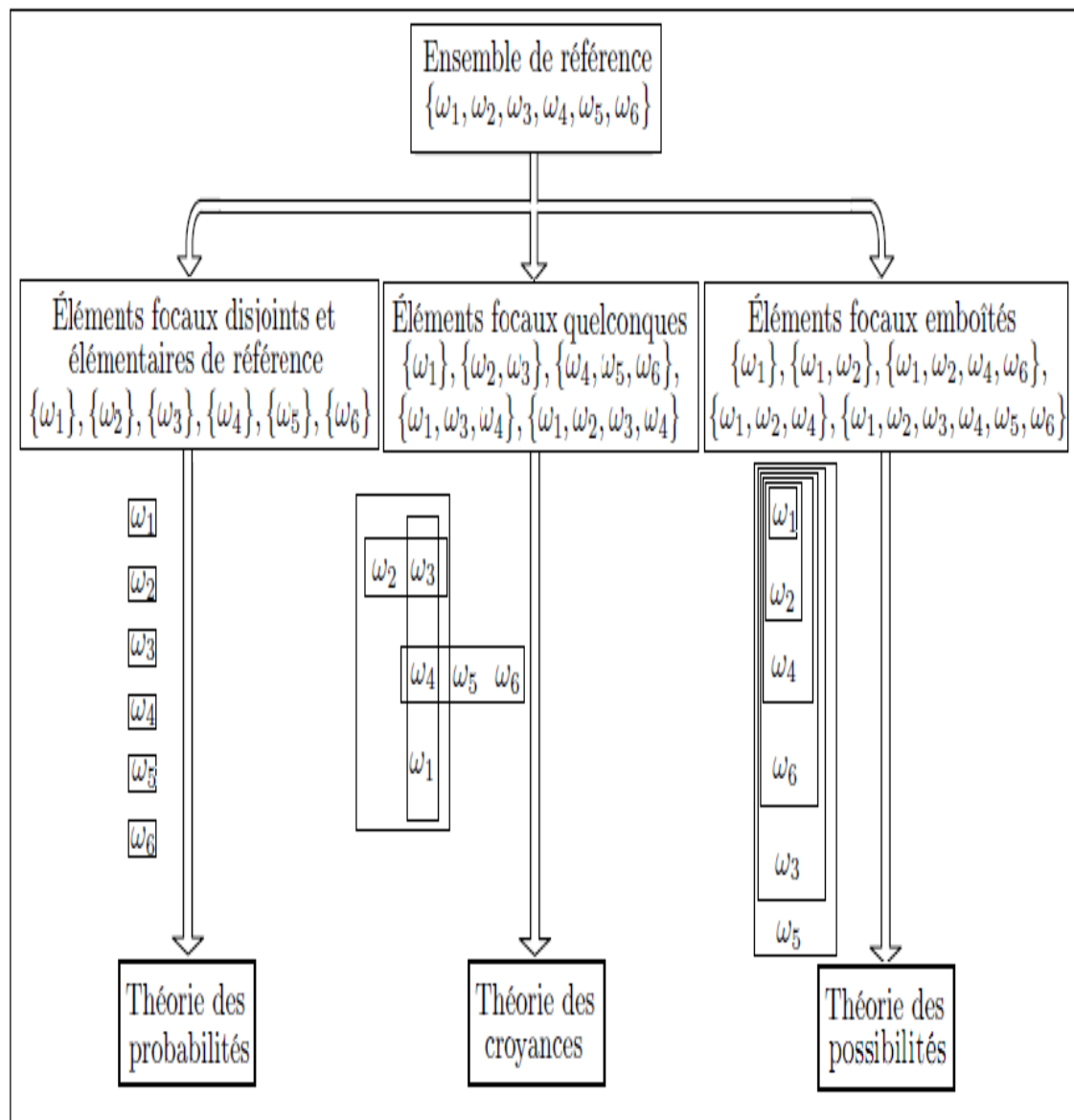


FIG. 3.7 – Position de la théorie des possibilités et la théorie des probabilités dans le cadre de la théorie de l'évidence d'après Bouchon-Meunier [7]

3.5 Relations de dominance et fonctions de croyance

Une fonction objectif d'un problème d'optimisation combinatoire déterministe est défini comme suit : $f : X \rightarrow Y \subseteq \mathbb{R}^n$

$$f(x) = \begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{pmatrix} = y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

où l'objectif est de trouver $x_{opt} \in X$ qui optimise (min ou max) f (FIG. 3.8, a). Sans perte de généralité, nous supposons par la suite que la fonction objectif est à minimiser.

Si $n = 1$, alors on traite un problème d'optimisation mono-objectif et la relation d'ordre sur Y est une relation d'ordre totale.

Si $n > 1$, alors il est nécessaire de définir la relation d'ordre au sens de Pareto \prec_p entre les vecteurs $y \in \mathbb{R}^n$ (Définition 1.2).

Mais quoi faire si les résultats de f sont des valeurs incertaines, représentées comme des intervalles des fonctions de croyance et de plausibilité (FIG. 3.8, b)? Formellement la fonction objectif devient : $f : X \rightarrow Y \subseteq \mathbb{R}^n$

$$f(x) = \begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{pmatrix} = y = \begin{pmatrix} [y_1, \bar{y}_1] \\ \vdots \\ [y_n, \bar{y}_n] \end{pmatrix}$$

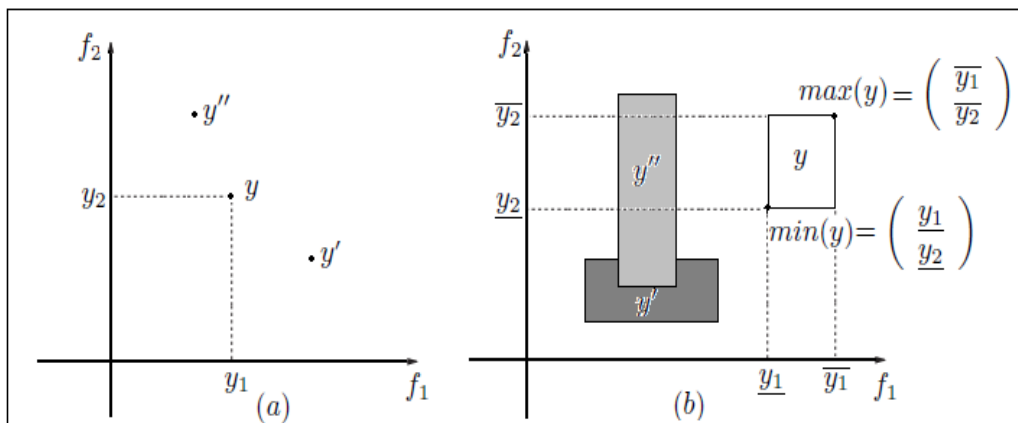


FIG. 3.8 – Valeurs de f (a) certaines et (b) incertaines.

Dans le cas déterministe, $f(x) = y$ est une valeur exacte. Si on compare y avec un autre vecteur y' , l'expression $y = y'$ est soit vraie ou bien fausse. Dans le cas probabiliste, $f(x) = Y$ est une distribution de probabilité, et $P(Y = y') \in [0, 1]$. Dans le cas de la théorie d'évidence Dempster-Shafer, la probabilité est bornée, $f(x) = Y$ est une masse de croyance multidimensionnelle, et l'égalité avec un vecteur y' peut être interprétée par $Bel(Y = y') \in [0, 1]$, $Pl(Y = y') \in [0, 1]$.

Le concept de l'incertitude se traduit par l'application de tels intervalles depuis l'espace décisionnel vers l'espace objectif. La prise en compte de l'incertitude entraîne donc la comparaison d'intervalles. Le véritable challenge soulevé par ce type de problèmes peut donc se résumer à la comparaison deux à deux d'intervalles de vecteurs d'objectif, plutôt que de vecteurs objectifs uniques dans le cas déterministe.

Afin de prendre l'incertitude en compte par le biais d'intervalles de vecteurs objectifs, on propose de modifier la relation de dominance telle que la dominance de Pareto.

On peut définir les relations de dominance entre deux intervalles de la façon suivante :

Définition 3.12 (Dominance certaine) (FIG. 3.9)

Un intervalle $y = [\underline{y}, \bar{y}] \subseteq \mathbb{R}$ domine un autre intervalle $y' = [\underline{y}', \bar{y}'] \subseteq \mathbb{R}$ de façon certaine (et on note $y \prec_c y'$) si et seulement si :

$$\bar{y} < \underline{y}'$$

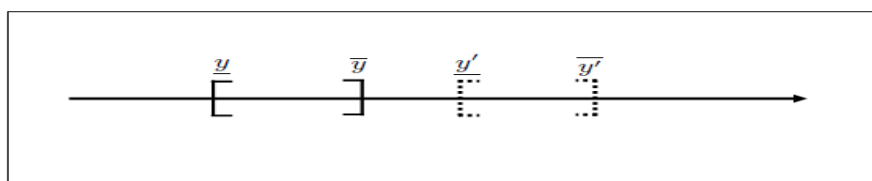


FIG. 3.9 – Dominance certaine ($y \prec_c y'$).

Définition 3.13 (Dominance incertaine) (FIG. 3.10)

Un intervalle $y = [\underline{y}, \bar{y}] \subseteq \mathbb{R}$ domine un autre intervalle $y' = [\underline{y}', \bar{y}'] \subseteq \mathbb{R}$ de façon incertaine (et on note $y \prec_{uc} y'$) si et seulement si :

$$\underline{y} \leq \underline{y}' \wedge \bar{y} \leq \bar{y}' \wedge y \neq y'$$

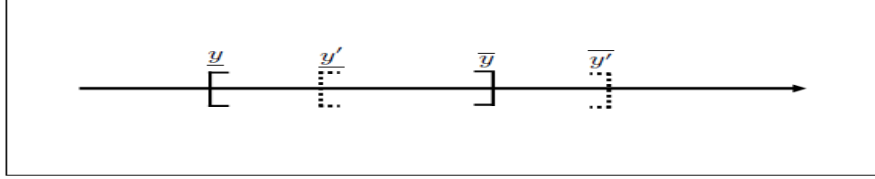


FIG. 3.10 – Dominance incertaine ($y \prec_{uc} y'$).

Définition 3.14 (Indifférence) (FIG. 3.11)

On dit que deux vecteurs y et y' sont indifférents (incomparables) (et on note $y \parallel y'$) si et seulement si :

$$\underline{y} \leq \underline{y}' \wedge \bar{y} \geq \bar{y}'$$

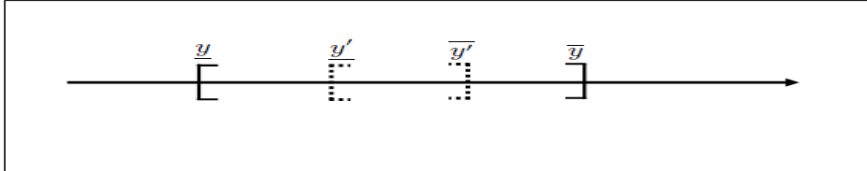


FIG. 3.11 – Indifférence ($y \parallel y'$).

Pour le cas des problèmes multi-objectif, on redéfinit la notion de dominance de Pareto comme suit.

Définition 3.15 (Dominance faible) (FIG. 3.12)

Un vecteur y domine faiblement un autre vecteur y' (et on note $y \prec_w y'$) si et seulement si :

$$\begin{aligned} \forall i \in 1 \dots n : y_i \prec^i y'_i \vee y_i \parallel y'_i \vee y_i = y'_i \\ \exists i \in 1 \dots n : y_i \prec^i y'_i \end{aligned}$$

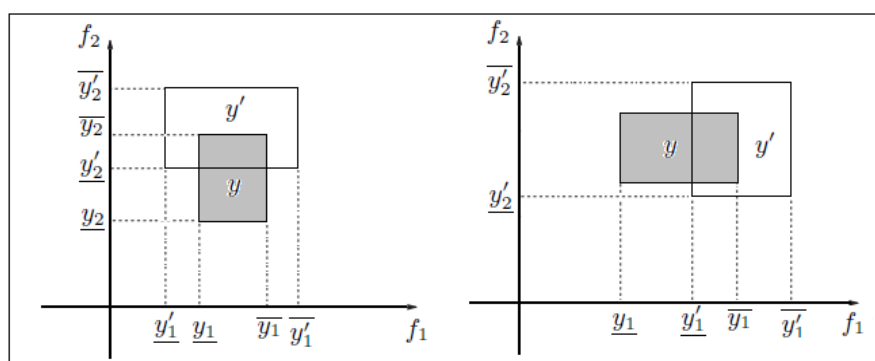


FIG. 3.12 – Dominance faible ($y \prec_w y'$).

Définition 3.16 (Dominance forte) (FIG. 3.13)

Un vecteur y domine fortement un autre vecteur y' (et on note $y \prec_s y'$) si et seulement si :

$$\forall i \in 1 \dots n : y_i \prec^i y'_i \vee y_i = y'_i$$

$$\exists i \in 1 \dots n : y_i \prec^i y'_i$$

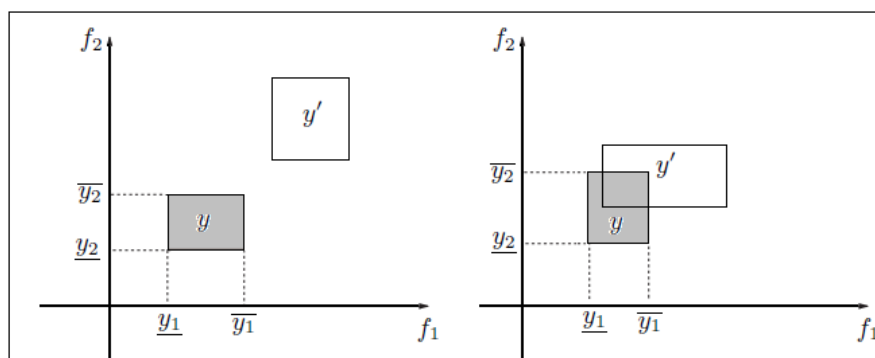


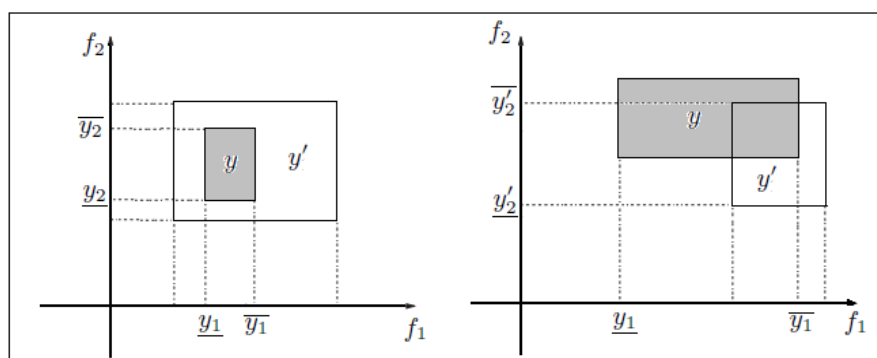
FIG. 3.13 – Dominance forte ($y \prec_s y'$).

Les deux relations se réduisent à la relation de dominance au sens de Pareto \prec_p lorsque $i = 1 \dots n : y_i$ et y'_i sont des intervalles dégénérés ($y_i = \bar{y}_i$) ou $i = 1 \dots n : \prec^i$ sont des ordres totales. La dominance forte se tient seulement si y est au moins égal à y' dans tous les objectifs et meilleur dans un seul tandis que la dominance faible tient si y est au moins indifférent de y' dans tous les objectifs et meilleur dans un seul.

Définition 3.17 (Indifférence) (FIG. 3.14)

On dit que deux vecteurs y et y' sont indifférents (incomparables) (et on note $y \parallel y'$) si et seulement si :

$$(\forall i \in 1 \dots n : y_i \parallel y'_i) \vee (\exists i, j \in 1 \dots n : y_i \prec^i y'_i \wedge y'_j \prec^j y_j, i \neq j, |I| = |J|, |I| + |J| = n)$$

FIG. 3.14 – Indifférence ($y \parallel y'$).

3.6 Conclusion

Dans ce chapitre, nous avons vu que l'imperfection dans les connaissances fait appel à trois concepts : l'imprécision, l'incertitude et l'incomplétude. Afin de gérer et de modéliser les imperfections, de nombreuses théories mathématiques existent dans la littérature. Les plus employées sont : la théorie des probabilités, la théorie des sous-ensembles flous, la théorie des possibilités, la théorie des fonctions de croyance. Plusieurs travaux consacrés à comparer ces théories mathématiques ont été réalisés dans la littérature. Par la suite, nous avons proposé de nouvelles relations de dominance pour la prise en compte de l'incertitude en nous basant sur la théorie des fonctions de croyance.

4

Approche Algorithmique

4.1 Introduction

Dans ce chapitre, nous présentons une approche algorithmique qui propose d'utiliser les nouvelles relations de dominance définies dans le chapitre précédent, pour la résolution du problème de tournées de véhicules avec demandes incertaines. Il concerne la conception, l'implémentation ainsi que l'analyse expérimentale de cette approche.

4.2 Algorithme DMLS

Dans l'algorithme DMLS (*Dominance-based Multiobjective Local Search*) une archive de taille fixe, bornée ou non bornée est utilisée comme stockage externe des solutions non dominées, et elle correspond aussi à la population à améliorer à l'aide des solutions voisines explorées, donc l'idée de base de cet algorithme consiste à maintenir une archive de solutions non dominées, d'explorer son voisinage et d'actualiser le contenu de cette même archive jusqu'à ce que celle-ci ne s'améliore plus.

Différentes variantes de cet algorithme ont été proposées dans la littérature, parmi lesquelles deux algorithmes de recherche locale Pareto PLS-1 et PLS-2 (*Pareto Local Search*) [60, 61], ainsi que PAES (*Pareto Archived Evolution Strategy*) [40] et moRBC($|A| : 1 + 1$)^A (*Random Bit Climber using the archive for restarts*) [2] sont d'autres exemples de cet algorithme. Or, ces quatre algorithmes partagent un grand nombre de composants communs. Un modèle conceptuel unifié, généralisant cette catégorie de métaheuristiques pour l'optimisation multi-objectif, est présenté dans la figure 4.1 et dans l'algorithme 1.

L'algorithme commence avec un ensemble de solutions ne se dominant pas les unes des autres. Ces solutions sont utilisées pour initialiser l'archive. Ensuite, trois étapes sont itérées jusqu'à ce qu'une condition d'arrêt soit satisfaite :

Étape 1 : un sous-ensemble de solutions de l'archive est choisi pour construire l'ensemble courant ;

Étape 2 : le voisinage de l'ensemble courant est examiné pour construire l'ensemble candidat ;

Étape 3 : l'archive est mise à jour avec les solutions de l'ensemble candidat en utilisant une relation de dominance.

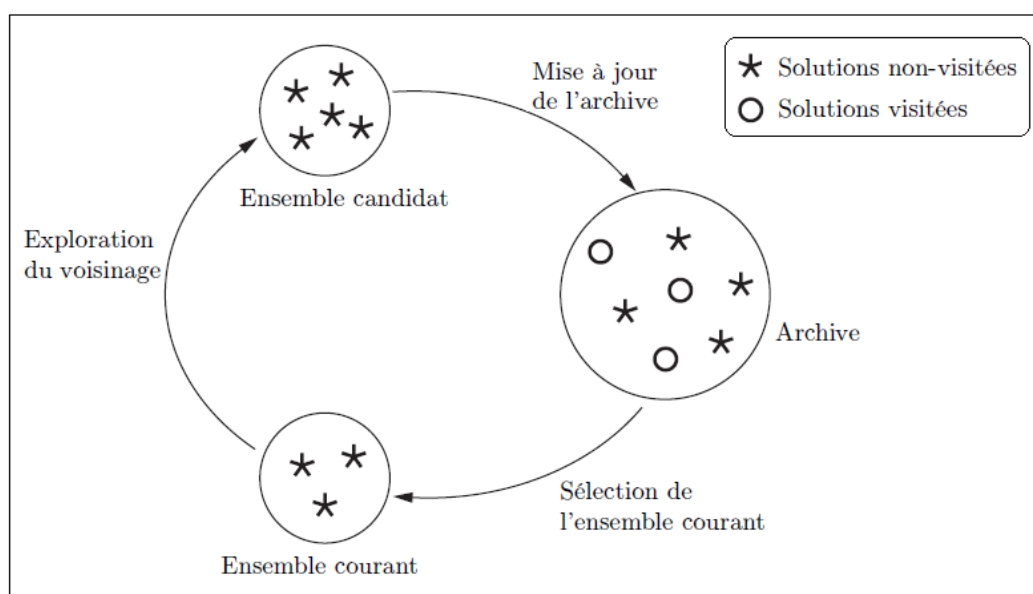


FIG. 4.1 – Étapes d'un algorithme DMLS.

Algorithme 1 DMLS

- 1: **Initialisation.** Démarrer avec un ensemble de solutions initiales P fourni en paramètres, ou le générer de façon aléatoire; initialiser l'archive A avec les solutions non-dominées de P .
 - 2: **Sélection de l'ensemble courant.** Sélectionner un ensemble de solutions non visitées de l'archive et ajoutez le à l'ensemble courant $P_{courant}$.
 - 3: **Exploration du voisinage.** Pour chaque solution x de l'ensemble courant $P_{courant}$, explorer le voisinage de x selon une certaine stratégie et ajouter les solutions non-dominées explorées à l'ensemble candidat $P_{candidat}$. Dans le cas d'une exploration exhaustive du voisinage, marquer x comme visitée.
 - 4: **Mise à jour de l'archive.** Mettre à jour l'archive A à l'aide des solutions de l'ensemble candidat $P_{candidat}$.
 - 5: **Condition d'arrêt.** Si les solutions de l'archive A sont toutes marquées comme visitées ou si une autre condition d'arrêt est satisfaite, retourner les solutions non-dominées de l'archive A . Stop. Sinon, aller à l'étape 2.
-

4.2.1 Description des composants

Les principaux composants permettant de concevoir un algorithme DMLS peuvent être définis comme suit :

Représentation. La représentation d'une solution est le point de départ de la conception de tout type de métaheuristique. Une solution doit être représentée aussi bien dans l'espace décisionnel que dans l'espace objectif. En effet, le choix de cette représentation constitue une étape de conception essentielle. De nombreuses représentations peuvent être appliquées telles que les représentations basées sur des vecteurs de valeurs réelles, discrètes, binaires ou des permutations.

Initialisation. La façon d'initialiser une solution est étroitement liée au problème à résoudre et à la représentation choisie. Dans la plupart des cas, la population initiale est générée de façon aléatoire. Cependant, certaines approches consistent à construire la solution initiale à l'aide d'une métaheuristique spécifique ou d'une méthode de construction.

Évaluation. Cette étape correspond au calcul et à l'affectation des valeurs objectif pour

une solution donnée. De ce fait, l'évaluation associe, à chaque solution réalisable, un vecteur de valeurs dont chaque élément quantifie la qualité de la solution considérée par rapport à la fonction objectif correspondante. Ainsi, il n'existe pas d'ordre total, mais uniquement un ordre partiel entre les solutions grâce à une relation de dominance. En pratique, l'évaluation est l'étape la plus coûteuse de la méthode en temps de calcul.

Structure de voisinage. La définition du voisinage est une question clé de l'efficacité de la recherche locale. Une fonction de voisinage N affecte un ensemble de solutions $N(x) \subset X$ à toute solution $x \in X$. Une solution $x' \in N(x)$ est appelée un voisin de x . Un voisin est le résultat de l'application d'un opérateur de mouvement m exerçant une légère perturbation à la solution x .

Évaluation incrémentale. Comme l'étape d'évaluation d'une solution est souvent l'étape la plus coûteuse d'un algorithme de recherche locale alors un moyen efficace d'évaluer une solution voisine est donc plus qu'appréciée. Cette évaluation incrémentale $\delta(x, m)$ va généralement dépendre de la solution courante x et du mouvement m à appliquer en vue de parvenir à la solution à évaluer.

Sélection de l'ensemble courant. La phase initiale d'une itération d'un algorithme DMLS porte sur la sélection d'un ensemble de solutions de l'archive à partir desquelles le voisinage est à explorer. En premier lieu, notons que, si certains membres de l'archive sont marqués comme visités, ils doivent être éliminés de l'ensemble sélection pour des raisons d'efficacité. En général, deux catégories principales peuvent être identifiées :

- *Sélection exhaustive*, où l'ensemble des solutions non-visités de l'archive est sélectionné ;
- *Sélection partielle*, où uniquement un sous-ensemble de solutions est sélectionné.

Exploration du voisinage. À partir de l'ensemble courant, un certain nombre de solutions candidates doivent être générées par le biais d'une structure de voisinage. Cet ensemble candidat est obtenu par une transformation locale répétée de chaque solution contenue dans l'ensemble courant. De même pour les stratégies de sélection de l'ensemble courant, deux catégories peuvent être distinguées pour une solution à explorer :

- *Exploration exhaustive du voisinage*, où le voisinage d'une solution est évalué de façon complète et déterministe ;

- *Exploration partielle du voisinage*, où seul un sous-ensemble de mouvements est appliqué.
 - *Voisin aléatoire* : Un seul voisin aléatoire de la solution actuelle est proposé comme candidat à l'intégration dans l'archive.
 - *Premier voisin non-dominé* : Pour chaque solution x de l'ensemble courant, le voisinage est évalué jusqu'à ce qu'un premier voisin non-dominé, à l'égard de x , soit trouvé.
 - *Premier voisin dominant* : Pour chaque solution courante x , les mouvements sont appliqués jusqu'à ce que l'on trouve une solution voisine dominant x . Ensuite, tous les voisins non-dominés évalués sont proposés comme solutions candidates à l'intégration au sein de l'archive.

Gestion de l'archive. La mise à jour du contenu de l'archive à l'aide de nouvelles solutions potentiellement non-dominées trouvées par l'algorithme repose généralement sur la relation de dominance Pareto. Mais d'autres types de relations peuvent être utilisés à la place de la dominance Pareto.

Condition d'arrêt. Les exemples les plus classiques consistent à stopper l'algorithme au bout d'un certain nombre d'itérations ou d'un certain temps d'exécution, etc. En effet, dans les algorithmes DMLS, lorsqu'il est possible de marquer les membres de l'archive comme visités, dépendamment de la stratégie d'exploration du voisinage choisie, un critère d'arrêt naturel consiste à poursuivre la recherche jusqu'à ce que toutes les solutions de l'archive soient marquées comme visitées. En effet, dans un tel cas, l'archive ne contient plus que des solutions non-dominées pour lesquelles tous les voisins sont dominés ou équivalent à au moins une solution de la même archive. L'algorithme est tombé dans une sorte d'optimum local Pareto.

4.2.2 Algorithmes existants

Le tableau suivant présente les algorithmes de recherche locale multi-objectif existants dans la littérature comme instances du modèle DMLS : PLS-1 (*Pareto Local Search*) [61] et PLS-2 [85], PAES (*Pareto Archived Evolution Strategy*) [40] et moRBC($|A| : 1 + 1$)^A (*Random Bit Climber using the archive for restarts*) [2].

Composants	PLS-1	PLS-2	PAES	moRBC($ A : 1 + 1)^A$
Relation de dominance	dominance Pareto	dominance Pareto	dominance Pareto	dominance Pareto
Sélection de l'ensemble courant	partielle 1 solution aléatoire	exhaustive toutes les solutions	partielle μ solutions	partielle 1 solution diversifiée
Exploration du voisinage	exhaustive tous les voisins	exhaustive tous les voisins	partielle λ voisins aléatoires	partielle 1 voisin dominant
Gestion de l'archive	non-bornée	non-bornée	bornée hyper-grille	bornée distance de crowding
Condition d'arrêt	naturelle solutions visitées	naturelle solutions visitées	définie par l'utilisateur	naturelle solutions visitées

4.3 Application au VRP avec demandes incertaines

Nous nous intéressons ici au 1-VRPTW avec demandes incertaines afin de trouver une solution qui minimise deux objectifs : la distance totale parcourue et le temps total de retard.

Formulation du problème

Les définitions et notations nécessaires pour la formulation du 1-VRPTW avec demandes incertaines sont données comme suit :

- *Clients et dépôt* : $V = \{0, 1, \dots, n\}$ représente les n clients à visiter. Chaque client v_i est géographiquement localisé aux coordonnées (x_i, y_i) . Le dépôt est représenté par le client 0, qui est le point de départ et d'arrivée pour chaque véhicule. Chaque client $v_i, i = 1, \dots, n$ a une demande incertaine représentée comme un intervalle $[d_i, \bar{d}_i]$. La demande réelle d_i est inconnue jusqu'à ce que le véhicule arrive chez le client. Une fenêtre de temps $[e_i, l_i]$ est associée à chaque client et au dépôt aussi, durant laquelle les clients doivent être servis et un temps de service s_i . Le dépôt a une demande $d_0 = 0$, et une fenêtre de temps $[0, l_0 \geq \max\{l_i : i \in V \setminus \{0\}\}]$.
- *Véhicules* : Un seul véhicule de capacité limitée Q est disponible pour servir les clients. Cette limite de capacité agit comme une contrainte et lorsque elle n'est pas satisfaite un échec de la route se produit. Un véhicule démarre son parcours du dépôt, il visite

un certain nombre de clients et il retourne au dépôt. Une route r est représentée par $r = (v_0, v_1, v_2, \dots, v_k, v_0)$ où k est la taille de la route.

- *Coûts euclidiens* : La distance de voyage d_{ij} entre deux points i et j , où chaque point peut être un client ou le dépôt, est égale au temps de voyage t_{ij} et on note c_{ij} , qui est calculé par l'équation suivante :

$$c_{ij} = d_{ij} = t_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

c_{ij} est symétrique ($c_{ij} = c_{ji}$), et satisfait l'inégalité triangulaire : $c_{ij} + c_{jk} \geq c_{ik}$.

- *Distance parcourue* : Dans ce problème, la demande réelle de chaque client n'est connue de façon certaine qu'à l'arrivée chez le client, lors du calcul de la distance sur une route $r = (v_0, v_1, \dots, v_f, \dots, v_k, v_0)$ définie a priori. Trois cas peuvent se présenter (Voir FIG. 4.2) :

1. Dans le cas où $\sum_{i=1}^f d_{v_i} < Q$, le véhicule servira le client f et passe au client suivant ($f + 1$).
2. Dans le cas où $\sum_{i=1}^f d_{v_i} = Q$ et $f < k$, le véhicule servira le client f , il retourne au dépôt pour recharger et il poursuivra la route prévue au client suivant ($f + 1$). Ces actions entraîneront des coûts supplémentaires de transport en terme de la distance parcourue $D(r)$ comme suit :

$$D(r) = \sum_{i=1}^{f-1} d_{v_i v_{i+1}} + \sum_{i=f+1}^{k-1} d_{v_i v_{i+1}} + d_{v_0 v_1} + d_{v_k v_0} + d_{v_f v_0} + d_{v_0 v_{f+1}}$$

3. Dans le cas où $\sum_{i=1}^f d_{v_i} > Q$, le véhicule servira le client f de ses capacités restantes ($Q - \sum_{i=1}^{f-1} d_{v_i}$), il retourne au dépôt pour recharger ensuite il retourne vers le même client f pour compléter son service (la quantité restante de sa demande) et finalement il continuera sa route au prochain client ($f + 1$). La distance totale parcourue associée à cette route est calculée de la façon suivante :

$$D(r) = \sum_{i=1}^{k-1} d_{v_i v_{i+1}} + d_{v_0 v_1} + d_{v_k v_0} + d_{v_f v_0} + d_{v_0 v_f}$$

- *Contraintes du temps* : En plus de définir les distances, les temps sont également nécessaires. Soient a_{v_i} , b_{v_i} les temps d'arrivée et de départ du client v_i respectivement. Le temps de départ du véhicule du dépôt est $b_{v_0} = 0$. Le temps d'arrivée chez le i ème client est calculé par :

$$a_{v_i} = b_{v_{i-1}} + t_{v_{i-1}v_i}$$

Lorsque le véhicule arrive tôt chez un client, il devra attendre le début de sa fenêtre de temps pour commencer son service, il y aura donc un temps d'attente W_{v_i} calculé comme suit :

$$W_{v_i} = \begin{cases} 0 & \text{si } a_{v_i} \geq e_{v_i} ; \\ e_{v_i} - a_{v_i} & \text{sinon.} \end{cases}$$

Ainsi, le temps de départ du i ème client est le suivant :

$$b_{v_i} = a_{v_i} + W_{v_i} + s_{v_i}$$

De même, en arrivant chez un client après la fin de sa fenêtre du temps, il y aura aussi un temps de retard T_{v_i} tel que :

$$T_{v_i} = \begin{cases} 0 & \text{si } a_{v_i} \leq l_{v_i} ; \\ a_{v_i} - l_{v_i} & \text{sinon.} \end{cases}$$

En cas d'échec d'une route, cela entraînera aussi des coûts supplémentaires de transport en termes de temps (FIG. 4.2).

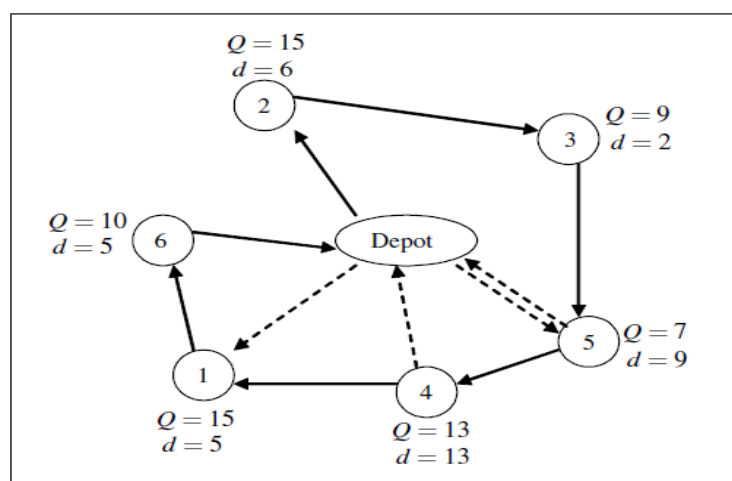


FIG. 4.2 – Exemple de modification de la route due au caractère incertain des demandes.

4.4 Implémentation sous ParadisEO

Pour développer l'algorithme de recherche local multi-objectif basé sur les relations de dominance des intervalles présentés précédemment, nous avons utilisé une plateforme dédiée à la conception de métaheuristiques pour l'optimisation multi-objectif : ParadisEO-MOEO [51].

4.4.1 Présentation de la plateforme ParadisEO

ParadisEO est une plateforme logicielle orientée-objet dédiée à la conception des métaheuristiques pour la résolution des problèmes d'optimisation de nature continue, discrète ou combinatoire. Basé sur la librairie EO (Evolving Objects), ParadisEO s'appuie sur la notion de Template C++, et il est portable sur les systèmes Unix (Linux, MacOS) et Windows. Il est composé des quatre modules suivants inter-connectés qui constituent une plateforme globale (FIG. 4.3) :

1. **ParadisEO-EO** pour l'implémentation de métaheuristiques à base de population, comprenant entre autres les algorithmes évolutionnaires et les algorithmes à essaims de particules.
2. **ParadisEO-MO** pour l'implémentation de métaheuristiques à base de solution unique, c'est-à-dire les méthodes de recherche locale, de recuit simulé, de recherche taboue, etc.
3. **ParadisEO-MOEO** pour l'implémentation de métaheuristiques dédiées à l'optimisation multi-objectif.
4. **ParadisEO-PEO** pour l'implémentation de métaheuristiques parallèles et distribuées.

Par ailleurs, la plateforme globale ParadisEO offre une vaste gamme de composants pour l'implémentation de métaheuristiques hybrides, grâce à la connexion existante entre ses différents modules. Dans notre travail, nous avons exclusivement concentré sur le module consacré à l'optimisation multi-objectif, à savoir ParadisEO-MOEO.

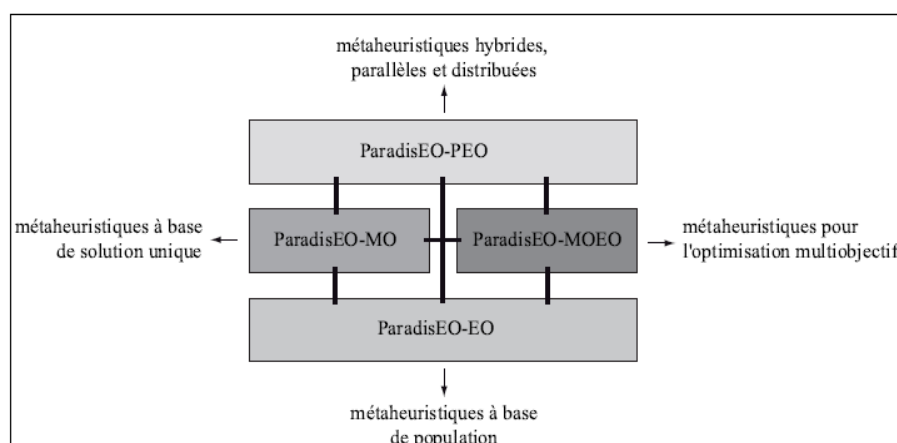


FIG. 4.3 – Modules de la plateforme ParadisEO.

4.4.2 Présentation du module ParadisEO-MOEO

ParadisEO-MOEO constitue une plateforme flexible et modulaire pour la conception et l'implémentation de métaheuristiques pour l'optimisation multi-objectif. À chaque niveau de son architecture, un ensemble de classes abstraites est proposé et un large éventail de classes instanciables, correspondant à différentes stratégies classiques de la littérature, est également fourni. En outre, la plateforme a pour but d'être extensible, souple et facilement adaptable, tous ses composants sont génériques afin que son architecture modulaire permet de développer rapidement et facilement toute nouvelle technique, ceci avec un minimum de lignes de code à écrire. L'objectif ici est de suivre les nouvelles stratégies à venir de la littérature et, si besoin est, de fournir des éléments supplémentaires nécessaires à leur mise en oeuvre. ParadisEO évolue constamment et de nouvelles fonctionnalités sont constamment susceptibles d'être ajoutées afin de fournir les concepts les plus modernes et efficaces, et de tenir compte des avancées les plus récentes du domaine.

Plusieurs classes fournies au sein de ParadisEO-MOEO permettent de définir un algorithme DMML de façon générique, par simple instantiation. Différents composants peuvent donc être très facilement expérimentés sans pour autant engendrer d'importantes modifications en termes d'écriture de code. Mais, il faut garder à l'esprit que cette liste n'est pas exhaustive, car la plateforme est en perpétuelle évolution et offre tout ce qui est nécessaire pour développer de nouveaux composants avec un minimum d'effort.

4.5 Analyse expérimentale

Les composants spécifiques liés à la résolution de notre problème à l'aide de métaheuristiques basées sur les algorithmes de recherche locale sont présentés ci-dessous.

Représentation. Le problème que nous considérons ici est un problème de VRP de permutation, i.e., une solution (chromosome) est représentée par une chaîne entière de longueur nombre de clients (n). Chaque gène dans ce chromosome indique l'identification d'un noeud assigné à l'origine à un client, tandis que l'ordre des gènes dans la chaîne du chromosome donne l'ordre de visite des clients.

Évaluation. L'étape d'évaluation se résume tout simplement au calcul des fonctions objectif liées au problème : la distance totale parcourue et le temps total de retard.

Initialisation. Les solutions initiales sont générées à l'aide de permutations aléatoires.

Voisinage. De nombreux opérateurs de voisinage existent pour des représentations à base de permutations, deux types d'opérateurs peuvent être distingués : les voisinages fondés sur la position des éléments et les voisinages fondés sur l'ordre des éléments. Les premiers se montrent peu adaptés à notre problème, pour lequel l'ordre des clients est plus important que leur position. Ainsi, un opérateur de voisinage, fondé sur l'ordre des éléments, et qui semble mieux approprié pour le problème de VRP étudié ici, s'avère être l'opérateur *2-Opt* qui consiste à échanger deux arcs (FIG. 4.4). C'est un opérateur couramment utilisé pour les problèmes de tournées et qui offre de bons résultats.

Évaluation incrémentale. Aucune évaluation incrémentale n'est considérée, chaque solution voisine sera évaluée à partir de zéro.

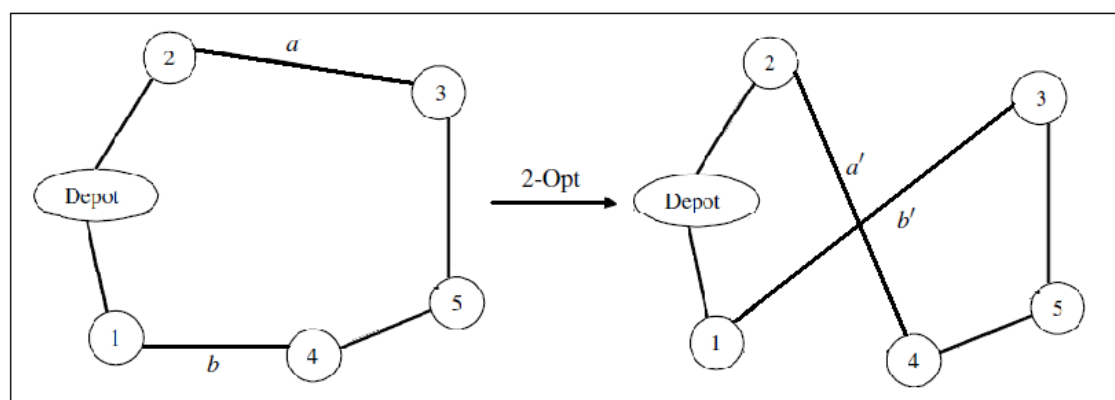


FIG. 4.4 – Opérateur de voisinage 2-Opt.

4.6 Design expérimental

L'algorithme utilisé est équivalent à PLS-1 [61] (voir Algorithme 2). À chaque itération, une seule solution aléatoire est sélectionnée parmi les solutions non visitées de l'archive, et explore son voisinage de façon exhaustive. Il contient une condition d'arrêt naturelle qui est vérifiée lorsque tous les membres de l'archive sont marqués comme visités. Le mécanisme d'archivage utilisé consiste à utiliser une archive non-bornée basée sur les relations de dominance des intervalles définies précédemment. L'algorithme est implémenté avec la version 1.3-beta de ParadisEO (le module ParadisEO-MOEO) fonctionnant sous Linux.

Algorithme 2 PLS-1

- 1: **Initialisation.** Initialiser l'archive A avec une solution $x \in X$ fournie en paramètre, ou la générer de façon aléatoire.
 - 2: **Sélection de l'ensemble courant.** Sélectionner une solution marquée comme non visitée x aléatoirement parmi les membres de de l'archive A .
 - 3: **Exploration du voisinage.** Générer la totalité des voisins de x et ajoutez les solutions non-dominées explorées à l'ensemble candidat $P_{candidat}$. Marquer x comme visitée.
 - 4: **Mise à jour de l'archive.** Mettre à jour l'archive A à l'aide des solutions de l'ensemble candidat $P_{candidat}$.
 - 5: **Condition d'arrêt.** Si les solutions de l'archive A sont toutes marquées comme visitées, retourner les solutions non-dominées de l'archive A . Sinon, aller à l'étape 2.
-

Dans le cadre de ce travail, pour tester notre approche, de nouvelles instances ont été développées à partir des instances de Solomon[79] : C1, C2, R1, R2, RC1, RC2.

Gardant les mêmes informations des problèmes de Solomon, sauf que les demandes sont représentées comme des intervalles générées de façon aléatoire. Six instances ont été testées : C101, C201, R101, R201, RC101, RC201 (voir exemple de l'instance C101 dans TAB. 4.1). Chacune de ces instances possède 100 clients et un seul dépôt. Ils renferment aussi les coordonnées de chaque client et le dépôt, leurs fenêtres de temps et le temps de service de chacun. Le nombre de véhicules a été fixé à 1 de capacité $Q = 60$. La taille de la population a été fixée expérimentalement à 500. Pour chaque instance, 30 exécutions ont été réalisées.

Client n°	Coord.X	Coord.Y	Demande	Temps au plus tôt	Temps au plus tard	Temps de service
0	40	50	0	0	1236	0
1	45	68	10	912	967	90
2	45	70	30	825	870	90
3	42	66	10	65	146	90
4	42	68	10	727	782	90
5	42	65	10	15	67	90
6	40	69	20	621	702	90
7	40	66	20	170	225	90
8	38	68	20	255	324	90
9	38	70	10	534	605	90
10	35	66	10	357	410	90
⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮
95	62	80	30	196	239	30
96	60	80	10	95	156	90
97	60	85	30	561	622	90
98	58	75	20	30	84	90
99	55	80	10	743	820	90
100	55	85	20	647	726	90

TAB. 4.1 – Instance C101.

4.7 Résultats

Les tableaux TAB. 4.2 et TAB. 4.3 résument les résultats obtenus pour les six instances testées dans les deux cas : déterministe et incertain. Chaque tableau montre la distance minimale et maximale, le retard minimal et maximal et le temps d'exécution en secondes pour chaque instance en utilisant l'algorithme de recherche local PLS-1 avec les nouvelles relations de dominance pour le cas incertain.

	Distance		Retard		Temps
	min	max	min	max	
C101	2333.98	2379.12	503916	527340	4.04
C201	2499.29	2620.91	461297	513502	4.71
R101	1979.98	2001.99	129575	141212	3.79
R201	2200.00	2317.95	119412	131314	4.32
RC101	2843.07	2926.05	166193	179236	3.85
RC201	2983.96	3082.96	145086	169361	4.00

TAB. 4.2 – Résultats du cas déterministe.

	Distance		Retard		Temps
	min	max	min	max	
C101	[2848.91,2896.82]	[2935.08,3021.06]	[538632,542652]	[579072,582057]	5.45
C201	[3343.18,3418.32]	[3531.25,3551.14]	[460409,461213]	[480645,483727]	5.92
R101	[2300.00,2340.01]	[2401.82,2508.76]	[140294,142477]	[150737,155332]	5.03
R201	[2522.47,2532.18]	[2704.40,2814.77]	[134144,135116]	[147427,147798]	5.52
RC101	[3484.85,3516.36]	[3600.91,3617.27]	[198381,199238]	[214301,217893]	5.32
RC201	[3438.18,3506.31]	[3611.36,3680.68]	[181536,184672]	[203037,205662]	5.80

TAB. 4.3 – Résultats du cas incertain.

On remarque que la valeur du retard total est considérable. Ceci est dû aux données fournies par les instances. Il y a plusieurs cas dans lesquels nous sommes dans l'obligation d'accuser un retard. Nous ne pouvons pas correctement comparer nos résultats avec d'autres à cause des incompatibilités entre les objectifs à optimiser. Les figures suivantes FIG. 4.5 à FIG. 4.16 montrent le front Pareto pour les deux cas déterministe et incertain (représenté comme des rectangles pour refléter les valeurs incertaines).

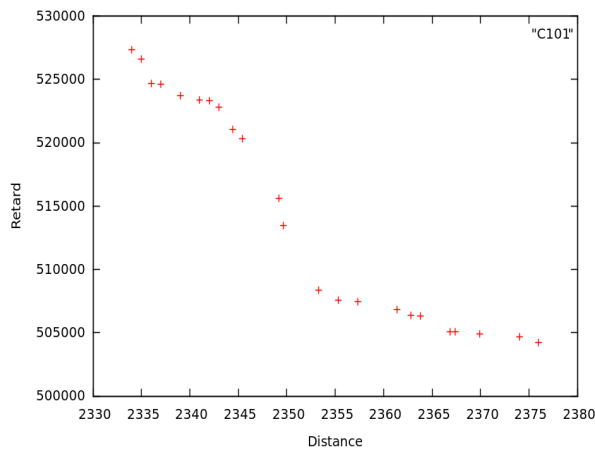


FIG. 4.5 - Cas déterministe (C101).

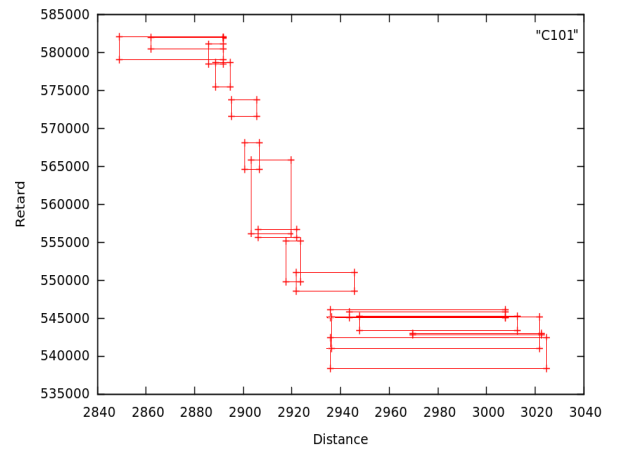


FIG. 4.6 - Cas incertain (C101).

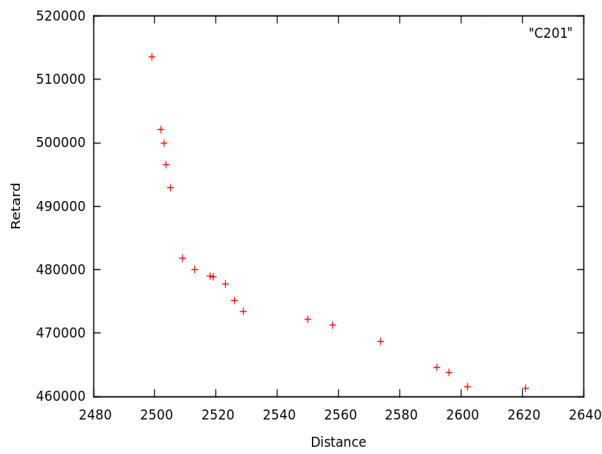


FIG. 4.7 - Cas déterministe (C201).

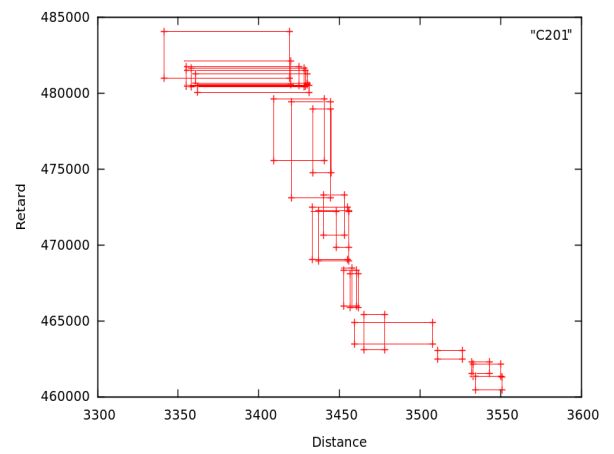


FIG. 4.8 - Cas incertain (C201).

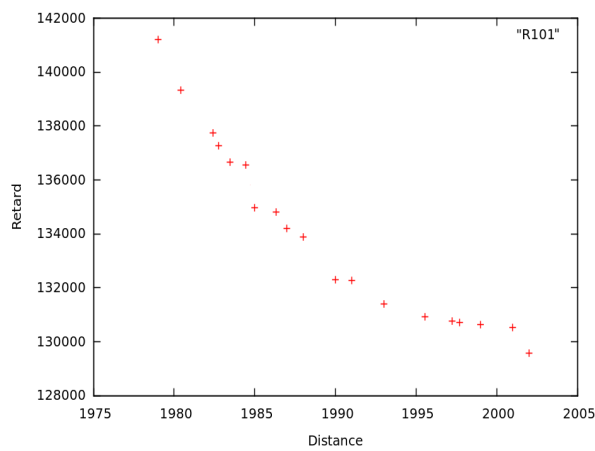


FIG. 4.9 - Cas déterministe (R101).

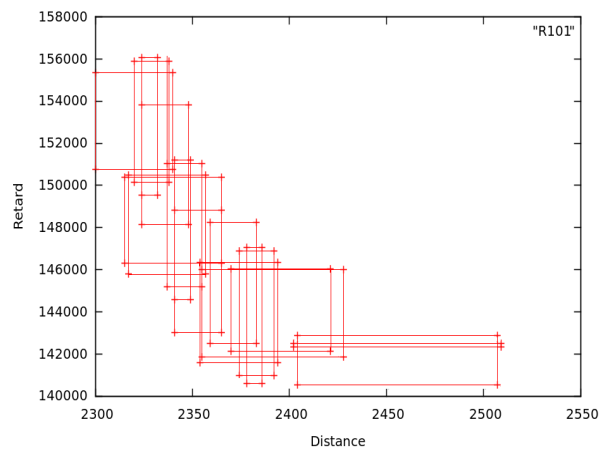


FIG. 4.10 - Cas incertain (R101).

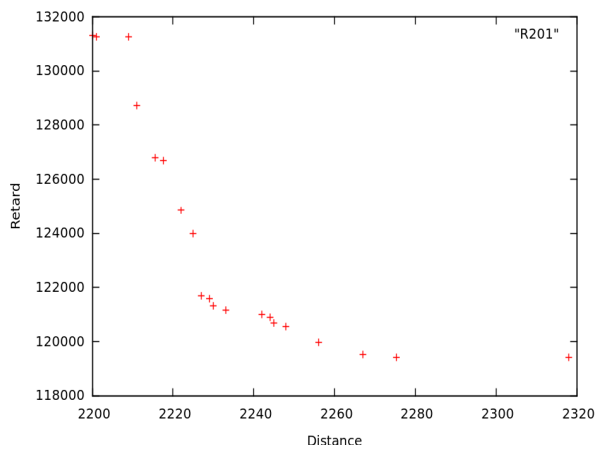


FIG. 4.11 - Cas déterministe (R201).

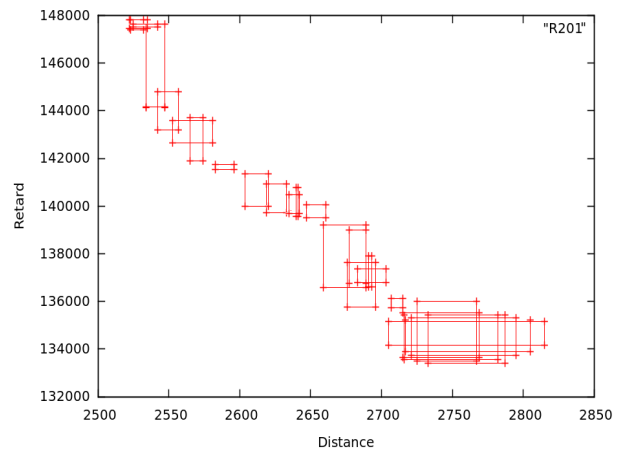


FIG. 4.12 - Cas incertain (R201).

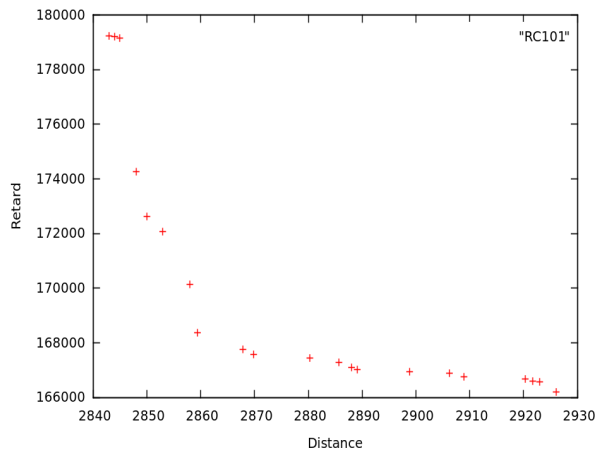


FIG. 4.13 - Cas déterministe (RC101).

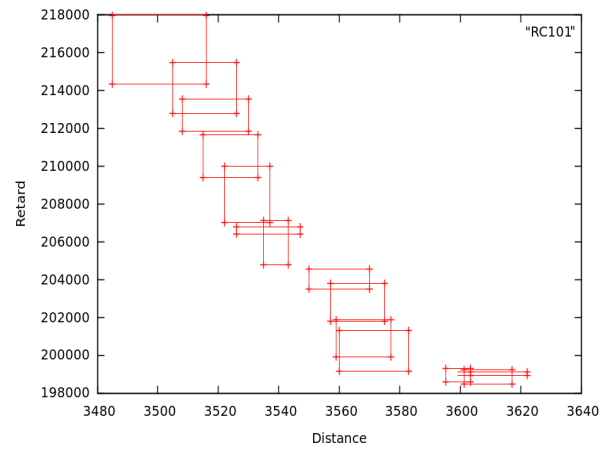


FIG. 4.14 - Cas incertain (RC101).

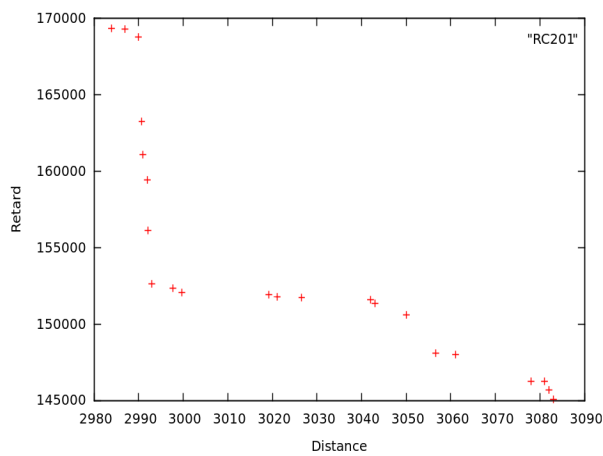


FIG. 4.15 - Cas déterministe (RC201).

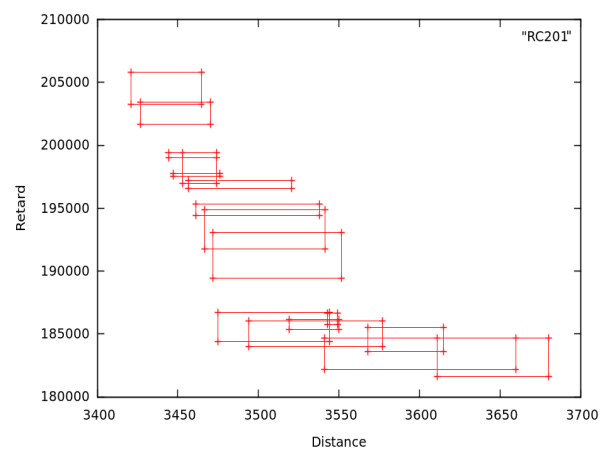


FIG. 4.16 - Cas incertain (RC201).

4.7.1 Analyse des résultats

À notre connaissance, il n'existe pas de méthodes adaptées à l'évaluation de performance d'algorithmes de recherche pour les problèmes d'optimisation multi-objectif avec incertitude. Par conséquent, nous proposons de réévaluer les solutions trouvées dans le cas déterministe sur les données incertaines et de comparer les fronts par rapport aux deux extrêmes fournis par les intervalles (PLS1-déterministe avec PLS1-incertain par rapport à Bel , et la même chose par rapport à Pl). Ainsi, il nous est possible d'utiliser des métriques classiques, employées dans le cas déterministe, pour mesurer la qualité d'un ensemble non-dominé, ou pour comparer deux approximations. La métrique que nous utilisons ici est l'*hypervolume* I_H [96].

- **Hypervolume** : cet indicateur mesure le volume de la portion multidimensionnelle de l'espace objectif faiblement dominée par une approximation donnée (FIG. 4.5). Il nécessite la spécification d'un point de référence f^{ref} . Ce dernier doit au moins être faiblement dominé par toutes les solutions de l'approximation considérée.

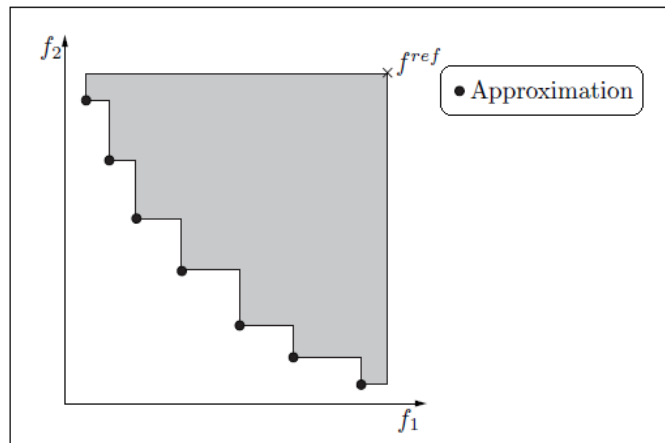


FIG. 4.5 – Indicateur hypervolume.

La méthode d'optimisation considérée est un algorithme stochastique. À chaque exécution, une approximation différente est retournée. Ainsi, nous obtenons un échantillon d'approximations correspondant à notre algorithme PLS : $(PLS_1, PLS_2, \dots, PLS_r)$ où r est le nombre d'exécutions de l'algorithme (dans notre cas $r = 30$). Ensuite relativement à l'indicateur d'hypervolume, nous transformons cet échantillon en un autre échantillon dont les éléments sont des valeurs scalaires : $(I_H(PLS_1), I_H(PLS_2), \dots, I_H(PLS_r))$. Enfin, nous résumons ces résultats à l'aide de statistiques descriptives telles que la moyenne de ces I -valeurs.

Le programme de calcul d'hypervolume est disponible sur la plateforme PISA [41]. Le point de référence que nous avons utilisé pour comparer les résultats observés sur une instance donnée est constitué des valeurs maximales obtenues lors de nos expérimentations pour chacun des critères sur cette instance.

Le tableau TAB. 4.4 fournit la comparaison de PLS1-déterministe (PLS1-dét) avec PLS1-incertain par rapport Bel (PLS1-inc (Bel)) et PLS1-incertain par rapport Pl (PLS1-inc (Pl)) selon la valeur moyenne de I_H .

Nous avons également appliqué une procédure de test statistique standard afin d'être en mesure de déclarer que "l'algorithme A surpasse l'algorithme B", "l'algorithme B surpasse l'algorithme A", ou "il n'existe pas de différence significative entre les algorithmes A et B".

		$I_H(\times 10^6)$	PLS1-dét	PLS1-inc (Bel)	PLS1-inc (Pl)
C101	PLS1-dét	0.671500	-	⋈	≡
	PLS1-inc (Bel)	1.938522	⋈	-	⋈
	PLS1-inc (Pl)	0.608895	≡	⋈	-
C201	PLS1-dét	4.858936	-	⋈	⋈
	PLS1-inc (Bel)	8.731948	⋈	-	⋈
	PLS1-inc (Pl)	2.493770	⋈	⋈	-
R101	PLS1-dét	0.169417	-	⋈	≡
	PLS1-inc (Bel)	1.150329	⋈	-	⋈
	PLS1-inc (Pl)	0.154230	≡	⋈	-
R201	PLS1-dét	1.761406	-	⋈	⋈
	PLS1-inc (Bel)	2.333240	⋈	-	⋈
	PLS1-inc (Pl)	1.102949	⋈	⋈	-
RC101	PLS1-dét	0.849234	-	⋈	≡
	PLS1-inc (Bel)	1.598426	⋈	-	⋈
	PLS1-inc (Pl)	0.848495	≡	⋈	-
RC201	PLS1-dét	1.712642	-	⋈	⋈
	PLS1-inc (Bel)	3.159762	⋈	-	⋈
	PLS1-inc (Pl)	1.545530	⋈	⋈	-

TAB. 4.4 – Comparaison des fronts selon l'indicateur d'hypervolume.

4.7.2 Discussion

Pour chaque instance, soit l'algorithme situé sur une ligne donnée domine significativement l'algorithme situé sur une colonne donnée (\succ), soit il est significativement dominé (\prec), soit il n'existe pas de différence significative entre les deux algorithmes (\equiv).

D'après les résultats obtenus d'hypervolume, le front PLS1-inc (*Bel*) de l'algorithme PLS1-incertain domine simultanément le front PLS1-inc (*Pl*) de l'algorithme PLS1-incertain et le front de l'algorithme PLS1-dét sur chacune des instances testées, ceci peut être expliqué par le fait que les demandes des clients sont les plus petites valeurs possibles à croire. De même pour le front de l'algorithme PLS1-dét qui domine le front PLS1-inc (*Pl*) de l'algorithme PLS1-incertain ou il n'existe pas de différence significative entre ces deux fronts, ceci peut être également expliqué par le fait que dans PLS1-inc (*Pl*) les demandes des clients sont les valeurs maximales les plus plausibles.

4.8 Conclusion

Dans ce chapitre, nous avons présenté la conception, l'implémentation et l'analyse expérimentale de l'algorithme de recherche locale multi-objectif basé sur une relation de dominance autre que la dominance de Pareto, pour la résolution du problème de tournées de véhicules bi-objectif avec des demandes incertaines.

Conclusion générale

Le travail effectué dans ce mémoire a porté sur la résolution de problèmes de tournées de véhicules multi-objectif avec des demandes incertaines.

Dans ce cadre, nous nous sommes intéressés au contexte multi-objectif en présentant les différentes métaheuristiques, et particulièrement les algorithmes de recherche locale.

Nous avons constaté que de nombreux problèmes d'optimisation multi-objectif sont caractérisés par la présence d'incertitude. Or, même si l'optimisation sous incertitude est raisonnablement étudiée dans le cas mono-objectif, ce domaine de recherche est très limité dans le contexte multi-objectif. Nous avons également observé que le concept de base de l'incertitude se traduisait par la comparaison d'ensembles depuis l'espace décisionnel vers l'espace objectif. Dans cette perspective, nous avons proposé d'utiliser de nouvelles notions de dominance pour la prise en compte de l'incertitude au lieu de la notion de dominance de Pareto et de les introduire dans un algorithme de recherche locale multi-objectif basé sur une relation de dominance.

Enfin, l'algorithme de recherche locale multi-objectif basé sur ces relations de dominance a été mis en pratique sur le problème de VRP avec demandes incertaines. L'algorithme a été développé dans une plateforme logicielle dédiée à l'implémentation des métaheuristiques multi-objectif : ParadisEO-MOEO. Celle-ci fournit un grand nombre de composants et mécanismes pour l'optimisation multi-objectif et ils peuvent être aisément utilisés ou modifiés et enrichis par de nouveaux composants de recherche. Par la suite nous avons obtenu des résultats encourageants.

Ce travail a permis non seulement la résolution dans le contexte multi-objectif du problème de tournées de véhicules sous incertitude, mais aussi de mieux connaître l'optimisation multi-objectif, les différentes théories pour la modélisation d'une information incertaine ainsi que les différentes caractéristiques du VRP. Il est très intéressant de poursuivre des recherches dans cette voie et un effort reste à fournir dans ce domaine. Comme perspectives à notre travail, nous proposons :

- La description du problème avec des données incertaines autres que les demandes, telles que les temps de voyage et l'existence d'un client.
- L'application de cette approche à d'autres problèmes d'optimisation combinatoire multi-objectif.
- Les relations de dominance que nous avons proposées peuvent facilement être applicables, ce qui pourrait, par exemple, donner lieu à une hybridation entre plusieurs méthodes de résolution, et améliorer considérablement les résultats.
- Une autre perspective concerne l'évaluation des algorithmes en contexte multi-objectif incertain.

Ainsi, dans le domaine de l'optimisation multi-objectif en environnement incertain, de nombreuses questions restent ouvertes et concernant toutes les étapes : la modélisation, la résolution et l'analyses de performances.

Bibliographie

- [1] N.R. Achuthan, L. Caccetta and S.P. Hill. *An Improved Branch-and-Cut Algorithm for the Capacitated Vehicle Routing Problem*. *Transportation Science* 37(2) : 153-169, 2003.
- [2] H. Aguirre and K. Tanaka. *Random bit climbers on multiobjective MNK-landscapes : Effects of memory and population climbing*. *IEICE Trans. Fundamentals*, 88-A(1) : 334-345, 2005.
- [3] B.M. Baker, M.A. Ayechev. *A genetic algorithm for the vehicle routing problem*. *Computers & OR* 30(5) : 787-800, 2003.
- [4] B. Baràn and M. Schaerer. *A multiobjective ant colony system for vehicle routing problem with time windows*. Dans *IASTED International Conference on Applied Informatics*, pp. 97–102, 2003.
- [5] J. Berger and M. Barkaoui. *A Hybrid Genetic Algorithm for the Capacitated Vehicle Routing Problem*. *GECCO* : 646-656, 2003.
- [6] L. Bianchi, M. Birattari, M. Chiarandini, M. Manfrin, M. Mastrolilli, L. Paquete, O. Rossi-Doria and T. Schiavinotto. *Metaheuristics for the vehicle routing problem with stochastic demands*. In S. B. . Heidelberg, editor, *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242, pages 450-460, 2004.
- [7] B. Bouchon-Meunier. *La logique floue et ses applications*, Addison-Wesley, 1995.
- [8] D.M. Chitty and M.L. Hernandez. *A hybrid ant colony optimisation technique for dynamic vehicle routing*. In : K. Deb et al. (Eds.), *GECCO 2004, Lecture Notes in Computer Science*, vol. 3102, Springer-Verlag, pp. 48–59, 2004.
- [9] A. Corberan, E. Fernandez, M. Laguna and R. Marti. *Heuristic solutions to the problem of routing school buses with multiple objectives*. *Journal of the Operational Research Society* 53, pp. 427–435, 2002.

-
- [10] D. Corne, J.D. Knowles and M.J. Oates. *The pareto envelope-based selection algorithm for multiobjective optimization*. In Conference on Parallel Problem Solving from Nature (PPSN VI), volume 1917 de Lecture Notes in Computer Science, pages 839-848, Londond, UK. Springer-Verlag, 2000.
- [11] G. Dantzig and J. Ramser. *The truck dispatching problem*. Management Science, 6 :80-91, 1959.
- [12] K. Deb, S. Agrawal, A. Pratap and T. Meyarivan. *A fast and elitist multiobjective genetic algorithm : NSGA-II*. IEEE Transactions on Evolutionar Computation. 6(2) :182-197, 2002.
- [13] A. Dempster. *Upper and lower probabilities induced by multivalued mapping*. Annals of Mathematical Statistics, vol. AMS-38, pp. 325-339, 1967.
- [14] D. Dubois et H. Prade. *Théorie des possibilités : applications à la représentation des connaissances en informatique*. Paris : Masson, 1985.
- [15] F. Y. Edgeworth. *Mathematical physics*. P.Keagan, Londres, Angleterre,1881.
- [16] M. Ehrgott. *Multicriteria Optimization*. Springer, Mars 2005.
- [17] N. El-Sherbeny. *Resolution of a vehicle routing problem with multi-objective simulated annealing method*. Thèse de doctorat, Faculté Polytechnique de Mons, Mons, Belgique, 2001.
- [18] C.M. Fonseca. *Multiobjective genetic algorithms with applications to control engineering problems*. PhD thesis, University of Sheffield, 1995.
- [19] C.M. Fonseca and P.J. Fleming. *Genetic algorithme for Multiobjective Optimisation : Formulation, Discussion and Generalization*. In Proceedings of the Fifth International Conference on Genetic Algorithms, San Mateo, California, p.416-423, 1993.
- [20] M.P. Fourman. *Compaction of Symbolic Layout using Genetic Algorithms*. In Genetic Algorithms and their Applications : Proceedings of the First International Conference on Genetic Algorithm, pp. 141-153, 1985.
- [21] X. Gandibleux, N. Mezdaoui and A. Freville. *A tabu search procedure to solve multiobjective combinatorial optimization problems*. In R. Caballero, F. Ruiz and R. Steuer editors, Second Int. Conf. on Multi-Objective Programming and Goal Programming MOPGP'96, pages 291–300, Torremolinos, Spain. Springer-Verlag, 1996.

-
- [22] M.J. Geiger. *Genetic algorithms for multiple objective vehicle routing*. Dans Metaheuristic International Conference 2001 (MIC'2001), pp. 348–353, 2001.
- [23] M. Gendreau, G. Laporte and R. Seguin. *An exact algorithm for the vehicle routing problem with stochastic demands and customers*. *Transportation science*, 29(2) :143-155, 1995.
- [24] D.E. Goldberg. *Genetic Algorithms in Search, Optimisation, and Machine Learning*. Addison-Wesley, Boston, MA, USA, 1989.
- [25] D.E. Goldberg and J. Richardson. *Genetic Algorithms with sharing for multimodal function optimization*. In 2nd International Conference on Genetic Algorithms and their application, pages 41-49, Mahwah, NJ, USA. Lawrence Erlbaum Associates, Inc, 1989.
- [26] M. Gronalt, R.F. Hartl and M. Reimann. *New Savings Based Algorithms for Time Constrained Pickup and Delivery of Full Truckloads*. *European Journal of Operational Research* 151 (3), 520-535, 2003.
- [27] J.H. Holland. *Adaptation in Natural and Artificial Systems*. university of Michigan Press, Ann Arbor, MI, USA, 1975.
- [28] J. Horn et N. Napfliotis. *Multiobjective Optimisation using the Niched Pareto Genetic Algorithm*. Illigal TR. n°93005, July 1993.
- [29] H.S. Hwang. *An Improved Model for Vehicle Routing Problem with Time Constraint Based on Genetic Algorithm*. *Computer Industrial Engineering*, (42) :361-369, 2002.
- [30] G. Ioannou, M. Kritikos and G. Prastacos. *A problem generator-solver heuristic for vehicle routing with soft time windows*. *OMEGA*, 31(1), 41-53, 2003.
- [31] N. Jozefowiez, F. Semet and E-G. Talbi. *Multi-objective vehicle routing problems*. In : *European Journal of Operational Research*, 2007.
- [32] N. Jozefowiez. *Modélisation et résolution approchées de problèmes de tournées multi-objectif*. Ph.D. thesis, Laboratoire d'Informatique Fondamentale de Lille, Université des Sciences et Technologies de Lille, Villeneuve d'Ascq, France, December 2004.
- [33] N. Jozefowiez, F. Semet and E-G. Talbi. *Parallel and hybrid models for multi-objective optimization : Application to the vehicle routing problem*, in : J.J. Merelo Guervos et al.(Eds.), *Parallel Problem Solving from Nature VII*, Lecture Notes in Computer Science, vol. 2439, Springer-Verlag, pp.271–280, 2002.

-
- [34] N. Jozefowiez, F.Semet and E-G.Talbi. *Applications of multi-objective evolutionary algorithm*. Advance in Natural Computation, chapter A multi-objective evolutionary algorithm for the covering tour problem, vol. 1, World Scientific, pp. 247–267, 2004.
- [35] N. Jozefowiez, F. Semet and E-G. Talbi. *The bi-objective covering tour problem*. Computers and Operations Research 34, pp.929–1942, 2007.
- [36] N. Jozefowiez, F. Semet and E-G. Talbi. *Enhancements of NSGA II and its application to the vehicle routing problem with route balancing*. in : E-G.Talbi, P. Liardet, P.Collet, E. Lutton, M. Schoenauer (Eds.), Artificial Evolution 2005, 7th International Conference (EA'2005), Lecture Notes in Computer Science, vol. 3871, Springer, pp.131–142, 2006.
- [37] N. Jozefowiez, F. Semet and E-G. Talbi. *Target Aiming Pareto Search and its application to the vehicle routing problem with route balancing*, Journal of Heuristics, 2006.
- [38] C.P. Keller. *Multiobjective routing through space and time : The MVP and TDVRP problems*. Thèse de doctorat, Department of Geography, University of Western Ontario, London, Ontario, Canada, 1985.
- [39] C.P. Keller and M. Goodchild. *The multiobjective vending problem : A generalization of the traveling salesman problem*. Environment and Planning B : Planning and Design 15, pp. 447–460, 1988.
- [40] J.D. Knowles and D. Corne. *Approximating the nondominated front using the Pareto archived evolution strategy*. Evolutionary Computation, 8(2) : 149–172, 2000.
- [41] J.D. Knowles, L. Thiele, and E. Zitzler. *A tutorial on the performance assessment of stochastic multi-objective optimizers*. In EMO 2005, 2005.
- [42] A.N. Kolmogorov. *Foundations of the theory of probability*. Chelsea Pub Co ; 2nd edition, 1960.
- [43] F. Kursawe. *A variant of evolution strategies for vector optimization*. In H.Schwefel and R.Manner editors, Parallel Problem Solving from Nature, volume 496 of Lecture Notes in Computer Science, pp.193–197, Berlin. Springer-Verlag, 1991.
- [44] P. Lacomme, C. Prins and M. Sevaux. *Multiobjective capacitated arc routing problem*. In C.M.Fonseca and al., editors, Evolutionary Multi-criterion Optimization, volume 2632 de Lecture Notes in Computer Science, pp.550-564. Springer-Verlag,2003.

-
- [45] P. Lacomme, C. Prins and M. Sevaux. *A genetic algorithm for a bi-objective capacitated arc routing problem*. Computers and Operations Research 33, pp.3473–3493, 2006.
- [46] H.C. Lau, M. Sim and K.M. Teo. *Vehicle routing problem with time windows and a limited number of vehicles*. European Journal of Operational Research, Volume 148, Number 3, pp. 559-569(11), 1 August 2003 .
- [47] T-R. Lee and J-H. Ueng. *A study of vehicle routing problem with load balancing*. International Journal of Physical Distribution and Logistics Management 29, pp. 646–648, 1998.
- [48] J. Lenstra and A. Rinnooy Kan. *complexity of vehicle routing and scheduling*. Networks, 11(2) :221-227, 1981.
- [49] H. Li and A. Lim. *Local search with annealing-like restarts to solve VRPTW*, European Journal of Operational Research, Volume 150, Issue 1, 115-127, 2003.
- [50] A. Liefoghe. *Métaheuristiques pour l’optimisation multi-objectif : Approches coopératives, prise en compte de l’incertitude et application en logistique*. Thèse de doctorat, Université Lille 1, 2009.
- [51] A. Liefoghe, M. Basseur, L. Jourdan and E-G. Talbi. *ParadisEO-MOEO : A framework for Evolutionary Multi-objective Optimization*, in S. Obayashi, C. Poloni, K. Deb (eds), Evolutionary Multi-Criterion Optimization, Fourth International Conference (EMO 2007), vol. 4403 of Lecture Notes in Computer Science, Springer-Verlag, Matsushima, Japan, p. 386-400, 2007.
- [52] A. Liefoghe, L. Jourdan and E-G. Talbi. *Metaheuristics and cooperative approaches for the bi-objective ring star problem*. Computers & Operations Research, vol. 37, n.6, pp.1033-1044, 2009.
- [53] M. Mourgaya. *The periodic vehicle routing problem : planning before routing*. Thèse de doctorat, Laboratoire de Mathématiques Appliquées de Bordeaux, Université de Bordeaux 1, Bordeaux, France, juillet 2004.
- [54] T. Murata and R. Itai. *Multi-objective vehicle routing problems using two-fold EMO algorithm to enhance solution similarity on non-dominated set*. in : C.A. Coello Coello, A.H. Aguirre, E. Zitzler (Eds.), Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005, Lecture Notes in Computer Science, vol. 3410, Springer, Guanajanto, Mexico, pp. 885–896, 2005.

-
- [55] T. Murata and R. Itai. *Local search in two-fold EMO algorithm to enhance solution similarity for multi-objective vehicle routing problems.* in : S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, T. Murata (Eds.), EMO 2007, vol. 4403, Springer, pp.201–215, 2007.
- [56] W. L. Oberkampf, J. C. Helton, C. A. Joslyn, S. F. Wojtkiewicz, and S. Ferson. *Challenge problems : uncertainty in system response given uncertain parameters.* Reliability Engineering System Safety, vol. 85, pp. 11-19, 2004.
- [57] B. Ombuki, B.J. Ross and F. Hanshar. *Multi-objective genetic algorithm for vehicle routing problem with time windows.* Applied Intelligence 24, pp.17–30, 2006.
- [58] A. Nagar, J. Haddock and S. Heragu. *Multiple and bicriteria scheduling : A literature survey.* European Journal of Operational Research, 81 :88–104, 1995.
- [59] J. Pacheco and R. Marti. *Tabu search for a multi-objective routing problem* Rapport technique TR09-2003, Université de Valance, 2003.
- [60] L. Paquete and T. Stützle. *A two-phase local search for the bi-objective traveling salesman problem.* In C.M.Fonseca et al., editors, Evolutionary Multi-criterion Optimization, volume 2632 de Lecture Notes in Computer Science. Springer-Verlag,2003.
- [61] L. Paquete, M. Chiarandini, and T. Stützle. *Pareto local optimum sets in the biobjective traveling salesman problem : An experimental study.* In Metaheuristics for Multiobjective Optimisation, chapter 7, pages 177–199. Springer-Verlag, 2004.
- [62] S.N. Parragh, K.F. Doerner and R.F. Hartl. *A survey on pickup and delivery problems Part I : Transportation between customers and depot.* February 19, 2008.
- [63] S.N. Parragh, K.F. Doerner and R.F. Hartl. *A survey on pickup and delivery problems Part I : Transportation between pickup and delivery locations.* April 16, 2008.
- [64] V. Pareto. *Cours d'économie politique.* F. Rouge, Lausanne, Suisse,1896.
- [65] Y. Park and C.Koelling. *A solution of vehicle routing problems in multiple objective environment.* Engineering Costs and Production Economics 10, pp.121–132, 1986.
- [66] Y. Park and C. Koelling. *An interactive computerized algorithm for multicriteria vehicle routing problems.* Computers and Industrial Engineering 16, pp.477–490, 1989.
- [67] C. Prins. *A simple and effective evolutionary algorithm for the VRP.* To appear in Computers and Operations Research. MIC 2001, p. 143-147 Porto, Portugal, 16-20/07/2001.

-
- [68] M. Rahoual, B. Kitoun, M. Mabed, V. Bachelet and F. Benameur. *Multicriteria genetic algorithms for the vehicle routing problem with time windows*. In : Metaheuristic International Conference (MIC'2001), pp. 527–532, 2001.
- [69] M. Rahoual, W. Djoukhdjoukh. *Métaheuristique hybride Pareto pour le problème de tournées de véhicules avec fenêtres horaires*. In : Evolution Artificielle 2003, pp. 380–395, 2003.
- [70] M. Reimann, K. Doerner and R.F Hartl. *Analyzing a unified Ant System for VRPs and some of its variations*. In : Raidl et al. (Eds.) : Applications of Evolutionary Computing, Springer LNCS 2611, Berlin/Heidelberg, pp 300 - 310, 2003.
- [71] J. Renaud, F.F Boctor. *A sweep-based algorithm for the fleet size and mix vehicle routing problem*. European Journal of Operational Research, 140, 3 : 618-628, 2002.
- [72] R. Ribeiro and H.R. Lourenço. *A multi-objective model for a multi-period distribution management problem*. In : Metaheuristic International Conference 2001 (MIC'2001), pp. 91–102, 2001.
- [73] C. Rego and C. Roucairol. *Le problème de tournées de véhicules : Etude et Résolution Approchée*. Technical Report, inria, Février 1994.
- [74] P. Serafini. *Simulated annealing for multiple objective optimization problems*. In Tenth Int. Conf. on Multiple Criteria Decision Making, pages 87–96, Taipei, 1992.
- [75] D. Schaffer. *Multiple Objective Optimisation with Vector Evaluated Genetic Algorithm*. In genetic algorithm and their Applications : Proceedings of the First International Conference on Genetic Algorithm, p.93-100, 1985.
- [76] T. Sen, M. Raiszadeh and P. Dileepan. *A branch and bound approach to the bicriterion scheduling problem involving total flowtime and range of lateness*. Management Science, 34(2) :254–260,1988.
- [77] W. Sessomboon, K. Watanabe, T. Irohara and K. Yoshimoto. *A study on multiobjective vehicle routing problem considering customer satisfaction with due-time(the creation of Pareto optimal solutions by hybrid genetic algorithm*. Transaction of the Japan Society of Mathematical Engineering, 1998.
- [78] G. Shafer. *A mathematical theory of evidence*, Princeton, University press, 1976.
- [79] M.M. Solomon. *Algorithms for the vehicle Routing and Scheduling Problem with Time Window Constraints*. Operations Research, 41, pp.469-488, 1987.

-
- [80] N. Srinivas and K. Deb. *Multiobjective optimisation using non-dominated sorting in genetic algorithms*. Evolutionary Computation 2(8), pp.221–248,1994.
- [81] B. Stewar and C. White. *Multiobjective a**. Journal of the ACM, 38(4) :775–814,1991.
- [82] C. Sutcliffe and J. Board. *Optimal solution of a vehicle routing problem : Transporting mentally handicapped adults to an adult training centre*. Journal of the Operational Research Society 41, pp.61–67, 1990.
- [83] E.D. Taillard. *Parallel Iterative Search Methods for Vehicle Routing Problems*. Networks 23, 661-673, 1993.
- [84] E-G. Talbi. *Metaheuristics : from design to implementation*. Wiley, 2009.
- [85] E-G. Talbi, M. Rahoual, M.H. Mabed and C.Dhaenens. *A hybrid evolutionary approach for multicriteria optimization problems : Application to the flow shop*. In First International Conference on Evolutionary Multi-criterion Optimization (EMO 2001), volume 1993 de Lecture Notes in computer Science, pages 416-428, Zurich, Switzerland. Springer-Verlag, 2001.
- [86] K.C. Tan, T.H. Lee, Y.H. Chew and L.H.Lee. *A multiobjective evolutionary algorithm for vehicle routing problems with time windows*. In IEEE Internatinal Conference on System, Man, and Cybernetics, pages 361-366. IEEE Service Center, 2003.
- [87] K.C. Tan, T.H. Chew and L.H. Lee. *A hybrid multiobjective evolutionary algorithm for solving truck and trailer vehicle routing problems*. European Journal of Operational Research 172, 855–885, 2006.
- [88] C.D. Tarantilis, C.T. Kiranoudis and V.S Vassiliadis. *A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem*. J. Oper. Res. Soc. 54, 65-71, 2003.
- [89] P. Toth and D. Vigo. *Models, relaxations and exact approaches for the capacitated vehicle routing problem*. Discrete Applied Mathematics, 123, 1-3 : 487-512, 2002.
- [90] E. Ulungu, J. Teghem, P. Fortemps and D. Tuyttens. *MOSA method : a tool for solving moco problems*. Journal of Multi-Criteria Decision Analysis, 8 :221-236,1999.
- [91] D. White. *The set of efficient solutions for multiple-objectives shortest path problems*. Computers and Operations Research, 9 :101–107,1982.
- [92] LA. Zadeh. *Fuzzy Sets*. Information and control ; vol. 8, pp.338-353, 1965.

- [93] L.A. Zadeh. *Fuzzy Sets as a basis for a theory of possibility*. Fuzzy Sets and Systems ; vol.1, pp.3-28, 1978.
- [94] E. Zitzler and S. Künzli. *Indicator-based selection in multiobjective search*. In Conference on Parallel Problem Solving from Nature (PPSN VIII), volume 3242 de Lecture Notes in Computer Science, pages 832-842, Birmingham, UK. Springer-Verlag, 2004.
- [95] E. Zitzler, M. Laumanns and L. Thiele. *SPEA2 : Improving the strength Pareto evolutionary algorithm*. TIK Report 103, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Zurich, Switzerland, 2001.
- [96] E. Zitzler and L. Thiele. *SPEA2 : Multiobjective evolutionary algorithms : A comparative case study and the strength Pareto approach*. IEEE Transactions on Evolutionary Computation, 3(4) :257-271, 1999.
- [97] K.G. Zografos and K.N. Androustopoulos. *A heuristic algorithm for solving hazardous material distribution problems*. European Journal of Operational Research 152, pp.507–519, 2004.

Résumé

Bien que les métaheuristiques soient couramment utilisées pour résoudre des problèmes multi-objectif d'une part, et des problèmes soumis à incertitude d'autre part, très peu de travaux ont été menés sur ces deux aspects simultanément. Par exemple, les problèmes de transport sont habituellement traités sous une forme mono-objectif déterministe, alors qu'ils sont clairement multi-objectif et qu'ils sont soumis à de nombreux facteurs d'incertitude.

Dans ce mémoire, nous nous intéressons aux méthodes de recherche locale multi-objectif basées sur une relation de dominance et pour la prise en compte de l'incertitude, nous avons défini de nouvelles relations de dominance entre des vecteurs d'intervalles au lieu des vecteurs précis. Nous appliquons ces nouvelles relations de dominance à un problème de tournées de véhicules de permutation bi-objectif avec demandes incertaines.

Les expérimentations menées montrent que l'algorithme de recherche locale multi-objectif permet de trouver une bonne approximation de l'ensemble Pareto pour des problèmes de tournées de véhicules avec demandes incertaines.

Mots clés : Optimisation multi-objectif - Incertitude - Métaheuristiques - Recherche locale multi-objectif - Dominance entre des intervalles - VRP - Demandes incertaines.

Abstract

Although metaheuristics are commonly used for solving multi-objective problems on one side, and problems subject to uncertainty on the other side, very few studies have investigated these two aspects simultaneously. For example, transportation problems are usually studied in a deterministic form with single objective, whereas they are clearly multi-objective and they are subject to a wide range of uncertainty.

In this work, we focus on dominance based multi-objective local search methods and to take account of uncertainty, we propose a new relations of dominance between imprecise vectors instead of precise vectors. We apply these new relations of dominance to a bi-objective permutation vehicle routing problem with uncertain demands.

The experiments conducted show that the multi-objective local search algorithm allows to find a good approximation of the Pareto set for the vehicle routing problems with uncertain demands.

Keywords : Multi-objective optimization - Uncertainty - Metaheuristics - Multi-objective local search - Dominance between intervals - VRP - Uncertain demands.
