



Contributions à la théorie de la décision algorithmique : optimisation multi-objectifs et optimisation dans l'incertain

Olivier Spanjaard

► To cite this version:

Olivier Spanjaard. Contributions à la théorie de la décision algorithmique : optimisation multi-objectifs et optimisation dans l'incertain. Intelligence artificielle [cs.AI]. UPMC - Paris 6 Sorbonne Universités, 2011. tel-01562260

HAL Id: tel-01562260

<https://theses.hal.science/tel-01562260>

Submitted on 13 Jul 2017

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

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

UNIVERSITE PIERRE ET MARIE CURIE
UFR D'INGÉNIERIE



Habilitation à Diriger des Recherches

mémoire de synthèse

***Contributions à la théorie de la décision algorithmique :
optimisation multi-objectifs et optimisation dans l'incertain***

OLIVIER SPANJAARD

Coordinateur: PATRICE PERNY
Professeur à l'Université Pierre et Marie Curie

Rapporteurs: MATTHIAS EHRGOTT
Professeur à l'Université d'Auckland

MICHEL GRABISCH
Professeur à l'Université Paris Panthéon-Sorbonne

JÉRÔME LANG
Directeur de recherche CNRS, Université Paris-Dauphine

Examineurs: EVRIPIDIS BAMPIS
Professeur à l'Université Pierre et Marie Curie

MARC PIRLOT
Professeur à l'Université de Mons

DANIEL VANDERPOOTEN
Professeur à l'Université Paris-Dauphine

Présentée et soutenue publiquement le 9 décembre 2011

Remerciements

C'est avec une certaine appréhension que j'entame la section qui sera vraisemblablement la plus lue de ce document, aussi connue comme LA section impossible à réussir : les remerciements. La section où la moindre approximation ne pardonne pas. La section qui, dans le meilleur des cas, ne vous fâchera pas avec l'ensemble de la communauté.

Avant d'entrer dans le vif du sujet, je tiens à faire observer au(x ?) lecteur(s ?) que les travaux de recherche menant à une habilitation couvrent une période de temps moins resserrée que ceux menant à une thèse. En conséquence, il devient encore plus délicat de mentionner toutes les personnes envers qui on se sent redevable. Aussi, je prie par avance ceux qui se sentiraient oubliés de ne pas en tirer de conséquences hâtives : ce n'est qu'un simple malentendu !

Je remercie Matthias Ehrgott, Michel Grabisch, Jérôme Lang, Evripidis Bampis, Marc Pirlot et Daniel Vanderpooten de leur participation au jury. Comme nombre d'autres missions incombant aux chercheurs, la tâche d'évaluation est réalisée à titre gracieux, au sens où elle n'est pas obligatoire et où la principale récompense est de satisfaire une éventuelle curiosité. Qu'ils voient ici exprimée ma gratitude. Soutenir devant un tel auditoire est un plaisir pour moi.

Je remercie Patrice Perny, directeur de thèse lorsque j'étais doctorant, puis directeur d'équipe après mon intégration à l'UPMC. Lorsque j'ai débuté ma thèse sous sa direction, je payais mon café du matin en Francs (était-ce prémonitoire ?), les M2 n'avaient pas encore été inventés et Patrice était un maître de conférences non habilité. Lors de nos premières soumissions communes, nous découpons à la main les articles pour les faire passer du format A4 au format letter, avant de nous précipiter chez FedEx aux premières lueurs de l'aube pour les adresser à l'*American Association for Artificial Intelligence*, qui ne s'appellait pas encore *Association for the Advancement of Artificial Intelligence*. C'est dire que cela fait aujourd'hui un peu de route parcourue, et si celle-ci fut légère, Patrice y est pour beaucoup.

Je remercie les deux doctorants que j'ai eu le plaisir d'encadrer, Gildas Jeantet et Charles Delort (dans l'ordre chronologique !). Une partie des résultats présentés dans ce document sont issus des travaux conduits ensemble. En recherche comme dans la vie, on n'est rien sans les autres. Et c'est quand même plus drôle les soirées de deadlines à plusieurs.

Je remercie les différents collègues et co-auteurs que j'ai côtoyés assidûment au gré des années, à l'intérieur et à l'extérieur du LIP6. Je ne me lancerai pas dans l'exercice périlleux de les nommer individuellement, mais ils se reconnaîtront : vous contribuez grandement à faire que ce travail n'en soit parfois plus vraiment un, ce qui n'a pas de prix.

Je conclurai enfin en remerciant ma famille pour son affection et son soutien.

Avant-propos

Ce document présente une synthèse de mes travaux depuis ma prise de fonction comme maître de conférences à l'Université Pierre et Marie Curie (UPMC). Les travaux menés durant ma thèse, ou directement dans le prolongement de celle-ci, ne seront donc pas présentés ici, à l'exception de certains d'entre eux qui préfiguraient bien de mes orientations futures. Ma thèse portait sur des approches qu'on pourrait qualifier d'algébriques pour faire de l'optimisation combinatoire fondée sur les préférences (Spanjaard, 2003) : il s'agissait d'identifier des propriétés-clés sur les préférences qui permettent d'étendre les algorithmes connus pour des problèmes classiques (plus court chemin, arbre couvrant minimum). Les recherches que j'ai conduites depuis sont de nature plus essentiellement algorithmiques. Elles sont complémentaires des précédentes dans la mesure où l'on cherche non plus à obtenir des résultats couvrant la plus large classe possible de préférences, mais au contraire à exploiter finement les propriétés spécifiques d'un modèle particulier dans un objectif d'efficacité algorithmique.

Le choix de se focaliser sur les travaux postérieurs à ma thèse a été dicté, d'une part, par la nécessité de se conformer au format des documents d'habilitation à l'UPMC, qui est de 40 pages maximum (objectif *non* atteint...), d'autre part par le souci d'une présentation unifiée de mes travaux de recherche, conjugué à la volonté de mettre en avant les aspects algorithmiques.

Dans l'entête d'un document scientifique, il est aujourd'hui d'usage d'énumérer les mots-clés qui caractérisent la nature de son contenu. Si je devais me plier à l'exercice pour l'ensemble de mes travaux récents, une liste possible de mots-clés serait la suivante : théorie de la décision, optimisation combinatoire, algorithmique, décision dans l'incertain, décision multi-objectifs, optimisation multi-objectifs, optimisation robuste, représentations graphiques, décision séquentielle dans l'incertain. Un graphe de dépendances entre ces mots-clés est représenté sur la figure 1. Un sommet représente bien sûr un mot-clé, et la présence d'un arc de x à y signifie que la thématique x est utile dans l'étude de la thématique y . On voit que les liens entre les différentes thématiques sont multiples, et qu'ils sont structurés de manière incontestablement non-linéaire. La rédaction d'un document de synthèse requiert pourtant une présentation linéaire des thématiques de recherche abordées (remarquons qu'il n'en aurait pas été de même pour un document hypertexte !). D'un point de vue algorithmique, il s'agit donc de réaliser ce qu'on appelle communément un *rangement topologique* des sommets du graphe, c'est-à-dire de déterminer un ordre linéaire sur les sommets qui soit compatible avec l'ordre partiel représenté par le graphe. L'ordre que nous avons adopté pour ce document est indiqué sur la figure. Le choix de cet ordre de présentation n'est bien évidemment pas arbitraire. En effet, ma communauté principale est celle des chercheurs en décision, et c'est donc naturellement des modèles décisionnels vers les algorithmes que j'orienterai mon discours.

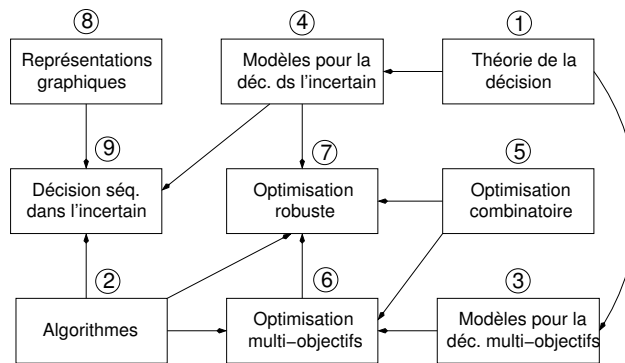


FIGURE 1 – Graphe de dépendances entre mots-clés.

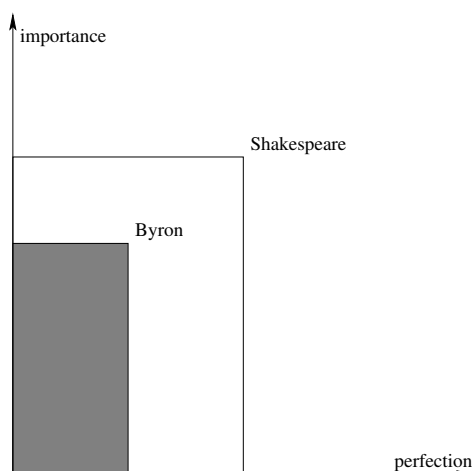
« Une fois que ma décision est prise, j'hésite longuement. »

Jules Renard.

1 Introduction

L'un des plus célèbres passages du film *Le cercle des poètes disparus* (scénario de Tom Schulman, réalisation de Peter Weir, 1989) met en scène le professeur Keating fustigeant J. Evans Pritchard, auteur imaginaire de l'ouvrage imaginaire *Comprendre la poésie*, détaillant dans sa préface la meilleure façon de comprendre et évaluer un poème.

« Afin de bien comprendre la poésie, on doit avant tout bien se familiariser avec la métrique, les rimes et les figures de style et ensuite se poser deux questions : 1) le thème du poème a-t-il été traité avec art et 2) ce thème est-il important. La première question définit la perfection de l'œuvre et la seconde son importance. Une fois que l'on a répondu à ces questions, il est relativement aisé de jauger la valeur de l'ouvrage. Si l'on note la perfection du poème sur la ligne horizontale d'un graphique, et son importance sur la verticale, l'aire totale ainsi obtenue par le poème indiquera la mesure exacte de sa valeur. Un sonnet écrit par Byron peut être noté haut sur la verticale mais avoir une côte médiocre sur l'horizontale. Un sonnet de Shakespeare en revanche sera noté très haut aussi bien à l'horizontale qu'à la verticale et la vaste surface couverte ainsi prouvera que la valeur du poème mérite tous les éloges. »



Ce qui est proposé ici, et que nous appelons *approche pritchardienne* dans la suite, est donc d'évaluer un poème sur la base de deux critères : la perfection et l'importance. Pour comparer deux poèmes, on utilise une fonction d'agrégation consistant à faire le produit des valeurs sur ces deux critères (ce qui revient à comparer les poèmes selon leurs moyennes géométriques). Cette méthode permet ainsi d'obtenir un classement des poèmes, dans une démarche qui pourrait s'apparenter à de l'*aide à la décision*. Toutefois, dans le cas présent, les propos de Keating semblent largement justifiés : « Il ne s'agit pas de mesurer de la tuyauterie. Il s'agit de poésie. Est-ce qu'on juge un poème comme dans le hit-parade mensuel ? » Faut-il pour autant condamner toute approche d'aide à la décision ? La réponse est bien évidemment négative de notre point de vue. Au-delà de l'arbitraire dans l'évaluation des critères et dans le choix du modèle de décision, une première piste pour appréhender ce qui dérange autant dans l'approche pritchardienne pourrait être de remettre en cause la nature des objets comparés. N'est-il pas dérisoire de vouloir introduire des éléments de rationalité dans un choix si étroitement lié aux affects ? L'activité d'aide à la décision s'est d'ailleurs initialement principalement focalisée sur des questions qui, bien que se posant concrètement dans l'industrie, sont de nature principalement mathématiques, à l'image de la programmation linéaire (Dantzig, 1947). Cette critique est cependant à relativiser car, avec le développement du web, les systèmes de recommandations se sont multipliés, qui visent à suggérer des musiques, des livres ou des films à des utilisateurs (voir figure 2), et qui relèvent clairement de l'aide à la décision au sens large.

En quoi ces systèmes d'aide à la décision diffèrent-ils de l'approche pritchardienne ? De par leur volonté affichée de tenir compte des préférences individuelles de l'utilisateur. En effet, une recommandation à un adolescent qui aime les films d'aventure et d'action différera fortement d'une autre qui s'adresserait à une retraitée fréquentant les cinémas d'art et essai. On peut toutefois reprocher à ces systèmes de ne pas tenir compte des attentes de l'utilisateur lors de la consultation. Autrement dit, ces systèmes répondent à une question qui n'a jamais été formulée en classant *dans l'absolu* des films sans tenir compte de ce que recherche l'utilisateur.



FIGURE 2 – Interface web d’un système de recommandation de films (allocine.com).

au moment présent (pour l’adolescent par exemple, s’agit-il d’aller voir un film avec ses amis ? avec sa dernière conquête féminine ? etc.). Ce qui distingue l’aide à la décision de l’approche pritchardienne tient ainsi principalement dans les points suivants :

- le problème de décision étudié est bien défini, c’est-à-dire que le décideur est clairement identifié et les attentes sont suffisamment cernées ;
- le choix du modèle de décision, ainsi que l’évaluation des objets comparés, se font en écartant autant que possible l’arbitraire ;
- la recommandation vise à déterminer une décision appropriée au regard du modèle de décision adopté.

Le respect de ces différents points donne une assise scientifique à l’activité d’aide à la décision, certains auteurs parlant même de *science de l’aide à la décision* (Roy et Bouyssou, 1993). Dans le cas contraire, les conclusions tirées de l’étude peuvent fortement prêter à discussion, comme l’illustre le débat actuel sur le classement de Shangaï, qu’on peut juger relever plus de l’approche pritchardienne que d’une véritable approche d’aide à la décision : le problème est mal structuré (que signifie « meilleure université du monde » ? quelles sont précisément les entités considérées dans le classement ?), le modèle de décision est frustré (une simple somme pondérée) et tient insuffisamment compte de l’hétérogénéité des critères, l’évaluation des différents critères comporte une part trop importante d’arbitraire (par exemple, un article a plus d’impact s’il est cosigné par de multiples auteurs)... Le lecteur intéressé pourra se reporter en particulier aux travaux de Billaut et al. (2010) sur le sujet.

Nos travaux s’inscrivent dans la vision scientifique de l’aide à la décision, dans le respect des différents points susmentionnés. Une fois le problème clairement identifié, on peut distinguer deux phases principales dans une intervention d’aide à la décision (Perny, 2000) :

- la *structuration* : concevoir un modèle qui soit une représentation de la réalité ;
- la *résolution* : explorer l’espace des possibles afin d’établir une recommandation.

Nos contributions portent essentiellement sur cette dernière phase. Lorsque l’espace des possibles est de nature combinatoire (composition d’un comité, localisation de services dans une agglomération), la détermination des alternatives préférées pose un problème algorithmique en

soi¹. L'avènement de la société de l'information a conduit à la multiplication des problèmes de décision impliquant un ensemble combinatoire d'alternatives potentielles : planification automatique d'itinéraires routiers (Goldberg et Harrelson, 2005; Bast et al., 2007), enchères combinatoires (Cramton, 2006), problèmes de configuration (Domshlak et al., 2001; Subbarayan et al., 2004)... L'abondance de problèmes de décision de ce type a favorisé l'émergence d'un nouveau domaine : la *théorie de la décision algorithmique*, qui mêle théorie de la décision et algorithmique discrète. C'est dans ce domaine que s'inscrivent nos travaux.

Le document est organisé comme suit. Après avoir présenté les principales problématiques abordées en théorie de la décision algorithmique (section 2), nous détaillons divers modèles décisionnels pour la décision multi-objectifs et la décision dans l'incertain (section 3). Nous décrivons ensuite nos contributions pour établir une recommandation selon ces modèles lorsque l'espace des alternatives est de nature combinatoire, aussi bien dans un contexte multi-objectifs (section 4) que dans l'incertain (section 5). Enfin, nous évoquons nos perspectives de recherche à court et moyen terme dans la conclusion (section 6).

2 Théorie de la décision algorithmique

Le terme *théorie de la décision algorithmique* est très récent. L'apparition de ce terme est principalement liée à la volonté d'une communauté de chercheurs en IA et en RO de promouvoir et d'identifier un ensemble de travaux mêlant théorie de la décision et algorithmique, suite à l'essor et à la popularisation de la *théorie des jeux algorithmique*, qui, elle, revisite la théorie des jeux sous l'angle de l'algorithmique (Nisan et al., 2007). Les préoccupations que recouvre l'intitulé *théorie de la décision algorithmique* sont cependant bien réelles et importantes. En effet, jusqu'à aujourd'hui, les multiples travaux en économie qui portent sur la théorie de la décision au sens large (y compris la théorie du choix social) se sont principalement focalisés sur la proposition de modèles et leur justification, en laissant au second plan les aspects computationnels. Autrement dit, l'*existence* de méthodes ou de solutions satisfaisant un certain nombre de conditions de rationalité prenait le pas sur les aspects *calculs*, d'autant plus facilement que la taille réduite des problèmes envisagés jusqu'alors autorisait une mise en œuvre pratique, même avec des représentations en machine et/ou des algorithmes non-optimisés. L'augmentation de la taille des problèmes rencontrés a induit de nouvelles problématiques qui sont devenues de plus en plus récurrentes, jusqu'à former le domaine de la théorie de la décision algorithmique que l'on connaît aujourd'hui. Sans prétendre à l'exhaustivité, nous nous efforçons d'en dresser un bref panorama dans la suite de cette section, en mettant en exergue les principales préoccupations de la théorie de la décision algorithmique.

La représentation des préférences. En amont de tout calcul en machine dans une optique d'aide à la décision, il est nécessaire de stocker les préférences de manière compacte lorsque l'ensemble des alternatives est de nature combinatoire, car stocker les préférences en extension devient rapidement prohibitif (en termes de ressources mémoires). Pour ce faire, on peut faire appel par exemple à des modèles graphiques, ou encore à des langages logiques de représentation des préférences.

A titre d'illustration, considérons les préférences du Chapelier fou d'*Alice au pays des merveilles* (Carroll, 1865). Lorsque celui-ci prend le thé, il aime accompagner sa boisson de chocolat

1. Remarquons que des difficultés algorithmiques peuvent survenir même lorsque l'ensemble des alternatives est de taille réduite. Par exemple, en décision multicritère, la mise en œuvre de la méthode ELECTRE requiert le calcul d'un *noyau* d'un graphe orienté (Roy et Bouyssou, 1993), c'est-à-dire un sous-ensemble S de sommets qui soit à la fois *stable* (absence d'arc entre les sommets de S) et *dominant* (pour tout sommet s' extérieur à S , il existe un arc d'un sommet s de S vers s'). Le problème consistant à décider s'il existe un noyau dans un graphe a été prouvé NP-complet (Chvátal, 1973).

noir quand le café est sans sucre, et de chocolat blanc dans le cas contraire. De plus, il ne refuse pas un nuage de lait dans son thé mais le prend parfois clair. Ses préférences partielles peuvent être synthétisées de la façon suivante : $(amer, clair, noir) \succ (sucre, avec\ nuage, blanc) \succ (amer, clair, blanc) \succ (sucre, clair, noir) \succ (amer, avec\ nuage, blanc)$, où $x \succ x'$ indique une préférence de x sur x' (malgré sa folie, les préférences du Chapelier sont supposées transitives). Soient x_i la $i^{ème}$ composante du tuple x , et u_i une fonction d'utilité sur cette $i^{ème}$ composante. On se convaincra sans peine qu'une utilité additive, c'est-à-dire définie sous la forme $u(x) = u_1(x_1) + u_2(x_2) + u_3(x_3)$, n'est pas compatible avec de telles préférences. En effet, cela nécessiterait de vérifier simultanément $u_1(amer) > u_1(sucre)$ (puisque $(amer, clair, noir) \succ (sucre, clair, noir)$) et $u_1(sucre) > u_1(amer)$ (puisque $(sucre, avec\ nuage, blanc) \succ (amer, avec\ nuage, blanc)$), ce qui est impossible. Par contre, une fonction d'utilité additive généralisée de la forme $u(x) = u_{12}(x_1, x_2) + u_3(x_3)$ permet de reproduire fidèlement les préférences du Chapelier, en prenant par exemple $u_{12}(amer, clair) = 4$, $u_{12}(sucre, avec\ nuage) = 5$, $u_{12}(sucre, clair) = 1$, $u_{12}(amer, avec\ nuage) = 1$, $u_3(noir) = 3$ et $u_3(blanc) = 1$. Les utilités additives généralisées ont été introduite par Fishburn (1970), et les modèles graphiques et algorithmes pour les manipuler ont été proposés par Gonzales et Perny (2004), en s'inspirant des travaux menés sur la programmation dynamique non-sérielle (Bertelé et Brioschi, 1972) ainsi que de l'abondante littérature dédiée à l'inférence dans les réseaux bayésiens (Pearl, 1988). Notons que d'autres modèles graphiques de représentation des préférences existent, parmi lesquels les plus connus sont vraisemblablement les *CP-nets* et leurs variantes, qui ont la particularité de représenter des préférences qualitatives (Boutilier et al., 2004). Un mode alternatif de représentation (logique) des préférences du Chapelier pourrait être de stocker une base de buts, comme suggéré par Uckelman et al. (2009), par exemple en utilisant la base de buts suivante : $B = \{(noir, 4), (blanc, 2), (clair, 3), (avec\ nuage, 2), (sucre, avec\ nuage, 2), (amer \wedge noir, 2)\}$. Dans ce formalisme, le tuple $(amer, clair, noir)$ se voit attribuer le score $4 + 3 + 2 = 9$ car, lorsque l'on compare la formule logique $\varphi = amer \wedge clair \wedge noir$ et la base de buts B , on a $\varphi \models noir$ (but de valeur 4), $\varphi \models clair$ (but de valeur 3) et $\varphi \models amer \wedge noir$ (but de valeur 2).

La question naturelle que l'on peut alors se poser est la suivante : quelle mode de représentation des préférences est le plus indiqué ? Il n'y a pas de réponse universelle à cette question. En effet, l'intérêt d'un mode de représentation des préférences se juge selon plusieurs aspects, qui sont la compacité de la représentation obtenue, l'expressivité du langage, la complexité des protocoles d'élicitation des préférences² dans ce langage, ainsi que la complexité des algorithmes de requêtes. Par exemple, la représentation des préférences sous la forme d'une utilité additive a de bonnes vertus de compacité : l'espace mémoire requis pour représenter des préférences sur un produit cartésien $X = X_1 \times \dots \times X_k$, où les ensembles X_i sont de modalité 2, est de l'ordre de $O(k)$, alors qu'une représentation explicite requiert $O(2^k)$. Par contre, du point de vue de l'expressivité, on est assez contraint sur le spectre des préférences qu'on est susceptible de représenter : les préférences du Chapelier par exemple n'étaient pas compatibles avec une utilité additive.

L'agrégation des préférences. Cette problématique consiste à déterminer un rangement des alternatives synthétisant plusieurs rangements représentant différents points de vue, ou plus simplement à déterminer une « meilleure » alternative (on parle de *vainqueur* en choix social, lorsque les alternatives représentent des candidats). Ainsi, en décision collective, on dispose des préférences individuelles des différents agents, et l'on cherche à obtenir un rangement reflétant une préférence collective. De façon assez similaire, en décision multicritère, on

2. Précisons que l'élicitation des préférences est une question très fréquemment abordée en théorie de la décision algorithmique, de multiples travaux y étant entièrement consacrés (Chajewska et al., 2000; Boutilier, 2002; Mousseau, 2003; Boutilier et al., 2006; Sandholm et Boutilier, 2006), et qu'on aurait pu la faire figurer au même plan que les principales problématiques mentionnées dans cette section.

cherche à obtenir une préférence globale à partir des préférences partielles selon chacun des critères. Pour approfondir les liens entre ces deux cadres, le lecteur intéressé pourra se reporter aux travaux de Bouyssou et al. (2009).

Empruntant un exemple de Faliszewski et al. (2009), montrons en quoi la problématique de l'agrégation est susceptible de comporter une importante composante algorithmique. Au pays des merveilles, la Reine souhaiterait décapiter le Lièvre de mars, le Chapelier fou ou le Loir car l'un d'eux est coupable (peu importe le chef d'inculpation). Le jury du procès est composé de la Reine, de la Duchesse et d'Alice (horrifiée). La Reine exprime les préférences *Chapelier fou* $\succ$ *Lievre de mars* $\succ$ *Loir*, où $\succ$ signifie « dont la décapitation est préférable à celle de ». Les préférences de la Duchesse, qui s'exprime suite à la Reine, sont *Lievre de mars* $\succ$ *Loir* $\succ$ *Chapelier fou*. Enfin, après avoir pris connaissance des préférences de la Reine et de la Duchesse, Alice se prononce généreusement pour *Loir* $\succ$ *Chapelier fou* $\succ$ *Lievre de mars*. Le mode d'agrégation des préférences choisi par la Reine consiste à ranger les potentiels condamnés selon leurs scores de Dodgson (1876), notion introduite par le mathématicien britannique du même nom. Charles Lutwidge Dodgson exerça comme professeur au *Christ Church College* à Oxford. Ecrivain à ses heures, il est plus connu du grand public comme auteur d'*Alice au pays des merveilles*, sous le pseudonyme de Lewis Carroll, ce qui explique vraisemblablement le choix de la Reine. Le score de Dodgson est fondé sur la notion de *vainqueur de Condorcet*. En théorie du choix social, un vainqueur de Condorcet est un candidat qui battrait tout autre candidat dans une confrontation au scrutin majoritaire. Selon les préférences exprimées par les membres du jury dans l'exemple du procès, si le Chapelier fou et le Lièvre de mars étaient seuls sur le banc des accusés, alors c'est le Chapelier fou qui serait décapité par deux voix contre une. Par contre, si le Chapelier fou et le Loir partageaient le banc, c'est le Loir qui serait décapité. Enfin, entre le Lièvre de mars et le Loir, c'est le Loir qui serait décapité en priorité. On observe donc qu'il n'y a pas ici de vainqueur de Condorcet, et qu'il y a même un circuit dans la relation majoritaire. Le score de Dodgson consiste à assigner à chaque candidat un entier égal au nombre minimum de permutations de voisins dans les préférences individuelles qui conduirait à le rendre vainqueur de Condorcet. Par exemple, il suffirait de permuter le Loir et le Chapelier fou dans les préférences de la Duchesse pour rendre le Chapelier fou vainqueur de Condorcet (le terme *vainqueur* étant toutefois ici quelque peu trompeur, puisqu'il s'agit d'être conduit à l'échafaud). Le score de Dodgson du Chapelier fou est donc 1. Un raisonnement similaire permet de conclure que les scores du Lièvre de mars et du Loir sont également de 1. Par conséquent, on obtient un préordre complet « de synthèse » où les trois accusés sont à égalité, ce qui devrait inciter la Reine à épargner les trois malheureux (le but recherché par Alice).

Cette méthode d'agrégation permet de construire un préordre complet à partir des préférences individuelles exprimées. La question algorithmique qui se pose alors est : quelle est la complexité du problème visant à déterminer le(s) vainqueur(s) d'une élection selon le score de Dodgson ? Une autre question en amont de la précédente est : quelle est la complexité du problème consistant à calculer le score de Dodgson d'un candidat ? Dans un article fondateur, Bartholdi III et al. (1989) ont établi que ces deux problèmes sont NP-difficiles. Remarquons toutefois que ces problèmes deviennent polynomiaux pour un nombre fixé de candidats ou lorsque le score de Dodgson maximum est borné (Betzler et al., 2010). En outre, la source de difficulté algorithmique n'est pas nécessairement liée à la nature de la règle d'agrégation utilisée : elle peut également provenir de la structure combinatoire de l'ensemble des alternatives considérées. Ainsi, lorsque les alternatives correspondent aux éléments d'un produit cartésien, comme cela peut être le cas par exemple dans des problèmes de configuration, la représentation explicite des préférences individuelles n'est rapidement plus envisageable (comme indiqué plus haut) et il est alors nécessaire de faire appel à une représentation compacte. Dans ce contexte

(où les préférences individuelles sont définies en compréhension sous forme de représentations compactes), déterminer une meilleure alternative, ou encore les k meilleures alternatives, pose des difficultés algorithmiques même lorsque la règle d'agrégation utilisée semble simple (pour une synthèse sur le sujet, on pourra se reporter à l'article de Chevaleyre et al., 2008).

La manipulation des procédures de choix. Cette problématique relève essentiellement du *choix social computationnel*, discipline récente qui vise à revisiter la théorie du choix social sous l'angle de la théorie de la complexité et de l'algorithmique. Dans le développement de la théorie du choix social, les économistes qui ont proposé et étudié différentes procédures de vote se sont intéressés, entre autres propriétés, à vérifier qu'une règle de vote encourageait à exprimer sincèrement ses préférences. Autrement dit, il s'agit de s'assurer que le dernier votant, si il connaît les autres bulletins de vote, ne pourrait pas avoir intérêt à voter de façon stratégique (mentir sur ses préférences) de façon à obtenir un vainqueur plus conforme à ses aspirations que s'il s'était exprimé sincèrement. Ainsi, dans l'exemple précédent du procès, Alice avait tout intérêt à exprimer les préférences qu'elle a indiquées afin d'épargner les trois accusés, alors qu'en son for intérieur elle n'avait aucune préférence. Si la manipulation est ici éclairée et souhaitable, c'est évidemment rarement le cas, et la résistance aux manipulations est considérée comme une propriété souhaitable des procédures de vote.

Le résultat le plus connu sur la manipulation des procédures de vote est le théorème de Gibbard-Satterthwaite (Gibbard, 1973; Satterthwaite, 1975; Benoît, 2000), qui indique qu'à partir de trois candidats la seule procédure de vote surjective (c'est-à-dire telle que tout candidat peut gagner l'élection) et résistante aux manipulations est la dictature. L'approche des économistes pour contourner ce résultat négatif consiste à restreindre le domaine des préférences autorisées. Par exemple, on peut s'intéresser aux *préférences unimodales*. Les préférences des votants sont dites unimodales lorsqu'il existe un axe linéaire, commun à tous les votants, le long duquel on peut classer les candidats, et que les préférences d'un votant décroissent au fur et à mesure que l'on s'éloigne de son candidat préféré dans un sens ou dans l'autre le long de cet axe. C'est souvent le cas par exemple en politique avec l'axe « gauche-droite ». Lorsque les préférences sont unimodales, Black (1948) a prouvé qu'il existe toujours un vainqueur de Condorcet. Pour un nombre impair de votants avec des préférences unimodales, le vainqueur de Condorcet correspond à la médiane (selon l'ordre linéaire correspondant à l'axe commun) des candidats préférés des différents votants. Cette procédure, valable pour des préférences unimodales uniquement, présente l'intérêt de ne pas être manipulable. Dans le cadre de cette approche, un premier travail pour les informaticiens peut consister, étant donné un profil de votes (c'est-à-dire un ensemble de rangements des candidats, un rangement par votant), à déterminer si les préférences sont unimodales, autrement dit si il existe un axe avec lequel toutes les préférences exprimées sont compatibles. Il a été montré que ce problème est polynomial (Trick, 1989; Escoffier et al., 2008a).

Une autre manière de lutter contre la manipulation, proposée et étudiée par les informaticiens, et qui présente l'intérêt d'être applicable en toutes circonstances (pas de restriction des profils de votes), consiste à utiliser des procédures dont on a établi que la manipulation est NP-difficile. Ainsi, dans l'article fondateur de Bartholdi III et al. (1989), il est montré que la règle de Borda (pour m candidats, cette règle consiste à attribuer $m - 1$ points au candidat classé premier dans les préférences d'un votant, $m - 2$ points au candidat classé second, et ainsi de suite et pour chaque votant) est manipulable avec un algorithme glouton polynomial : il suffit simplement de classer premier le candidat que l'on souhaite faire élire, puis, pour chacun des rangs suivants, s'il existe un candidat qu'on peut classer à ce rang sans remettre en cause l'élection du candidat que l'on souhaite faire élire, alors on le classe et on passe au rang suivant, sinon on abandonne car il n'est pas possible de faire élire le candidat recherché. A

contrario, Bartholdi et Orlin (1991) ont montré que manipuler la forme suivante de pluralité itérée est NP-difficile : à chaque itération, on élimine le candidat qui a le moins de voix à la pluralité (et les voix pour ce candidat sont « transférées » aux candidats suivants dans les votes concernés), et le dernier candidat restant est le vainqueur. Toutefois, la manipulation de cette procédure de vote devient un problème polynomial lorsque le nombre de candidats est fixé (Conitzer et al., 2007). Pour une introduction plus poussée aux thématiques abordées en choix social computationnel, le lecteur intéressé pourra se référer en particulier au tutoriel d’Ulle Endriss, disponible en ligne sur sa page web, ou encore à l’article introductif de Chevaleyre et al. (2007).

L’optimisation en présence de différents points de vue. Il est fréquent aujourd’hui de rencontrer des problèmes de choix où l’ensemble des alternatives potentielles est défini en compréhension et où l’on cherche à tenir compte de plusieurs points de vue dans la décision (points de vue qui peuvent correspondre à différents objectifs, scénarios ou encore agents). Déterminer une alternative qui optimise un critère de décision donné est alors souvent un problème d’optimisation délicat. Sans prétendre à l’exhaustivité, nous donnons maintenant quelques exemples de problèmes entrant dans cette catégorie.

L’optimisation combinatoire multi-objectifs (Ehrgott, 2005) en fait bien évidemment partie : il s’agit de trouver, parmi un ensemble combinatoire de solutions potentielles évaluées selon plusieurs objectifs, les solutions Pareto-optimales, ou une solution de meilleur compromis. Par exemple, la recherche automatique d’un itinéraire routier est un problème d’optimisation multi-objectifs, dès lors qu’on ne se contente pas de chercher le plus court chemin, mais que l’on tient également compte d’autres critères que la distance, tels que les coûts de péage. La plupart des applications commerciales actuelles semblent opter pour une optimisation lexicographique, c’est-à-dire qu’elles établissent un ordre d’importance des critères et cherchent d’abord à optimiser le premier critère, puis départagent les solutions optimisant ce critère en considérant le second critère, et ainsi de suite. L’optimisation lexicographique présente l’intérêt d’être facile à mettre en œuvre car les algorithmes mono-objectifs classiques sont aisément adaptables à ce modèle décisionnel. Cette approche est toutefois assez frustrante d’un point de vue décisionnel, ce qui justifie l’étude d’algorithmes multi-objectifs dédiés à d’autres modèles décisionnels.

Un autre type de problèmes mêlant décision et combinatoire est celui des problèmes de localisation d’installation(s) sur un réseau, tels que la localisation d’arrêts de bus ou d’entrepôts dans une ville, ou encore de *hubs* dans un réseau d’ordinateurs... L’objectif est alors de positionner la ou les installation(s) de telle sorte à satisfaire au mieux les usagers du réseau, qui sont eux-mêmes situés en différents points dudit réseau. La satisfaction d’un usager est généralement inversement proportionnelle à sa distance à l’installation la plus proche. Selon le critère utilisé pour agréger les satisfactions individuelles, le problème combinatoire induit peut changer assez profondément de nature et de difficulté. Parmi les critères proposés, mentionnons la somme, le max, la moyenne, la moyenne ordonnée pondérée (somme pondérée des distances, avec un poids plus important sur les plus grandes distances), et bien d’autres encore. L’étude de ce type de problèmes a donné lieu à une très abondante littérature (Current et al., 2001; ReVelle et Eiselt, 2005; Alumur et Kara, 2008), ainsi qu’à des revues dédiées (comme par exemple *Location Science*, aujourd’hui incorporée dans *Computers & Operations Research*). L’omniprésence des réseaux dans notre société moderne rend cette problématique particulièrement critique de nos jours.

Une autre catégorie beaucoup étudiée de problèmes est celle des problèmes d’affectation, ou plus largement de mise en relation d’offres et de demandes sur un marché « biface ». L’exemple le plus illustre est vraisemblablement le problème de mariage formalisé par Gale et Shapley

(1962). Etant donnés n hommes et n femmes, chaque homme exprime ses préférences sur les femmes à marier sous forme d'un rangement, et inversement chaque femme fait de même pour les hommes. Il s'agit ensuite de trouver un ensemble *stable* de mariages, c'est-à-dire tel qu'il n'existe pas un homme et une femme qui se préfère l'un et l'autre à leur compagne et leur compagnon actuels. Gale et Shapley (1962) ont proposé un mécanisme qui garantit de trouver un tel ensemble de mariages, en favorisant soit les femmes soit les hommes dans leurs choix. Au-delà de cet habillage ludique, ce problème ou des problèmes voisins sont très importants dans des applications concrètes telles que les campagnes d'admissions ou de recrutements, comme cela a été souligné par les auteurs eux-mêmes. A titre d'illustration, lors de la campagne 2009 de recrutement d'enseignants-chercheurs en France, 2216 postes de maîtres de conférences (resp. 1317 postes de professeurs) étaient au concours pour 9430 candidats (resp. 2159), chaque université ayant exprimé ses préférences via des comités de sélection, et chaque candidat exprimant ensuite ses préférences entre universités sur un serveur centralisé. Décider quelle affectation stable choisir est un problème éminemment combinatoire, et le choix doit être fait en respectant l'équité entre les candidats. Cela explique que les travaux menés sur le sujet impliquent à la fois des spécialistes en décision et des chercheurs en optimisation combinatoire (Baïou et Balinski, 2003; Haeringer et Iehlé, 2010). Les récentes études sur des problèmes d'affectations de papiers à des relecteurs s'inscrivent dans cette même lignée de travaux autour du problème de mariage stable (Garg et al., 2010; Lesca et Perny, 2010), avec cette fois-ci une priorité du bien-être social sur le bien-être individuel.

Pour terminer, parmi les travaux relevant de l'optimisation en théorie de la décision algorithmique, mentionnons la décision séquentielle dans l'incertain, qui traite de problèmes où l'on ne prend pas une unique décision, mais où l'on suit une *stratégie* (i.e. une séquence de décisions conditionnellement à des événements incertains) conduisant à une conséquence finale incertaine. Ce type de problème se rencontre en particulier en planification dans l'incertain en intelligence artificielle, comme par exemple pour la navigation d'un robot mobile, le diagnostic médical, l'organisation des secours ou encore la planification des actions d'un joueur artificiel. L'utilisation d'outils issus de la théorie de la décision dans ce contexte a donné naissance à un champ de recherche intitulé *decision-theoretic planning* (Blythe, 1999; Boutilier et al., 1999). L'objectif est bien évidemment de développer des algorithmes d'optimisation afin d'identifier un plan optimal au sens d'un modèle décisionnel donné.

Au sein de la théorie de la décision algorithmique, nos travaux portent principalement sur la thématique d'optimisation en présence de différents points de vue, et plus particulièrement sur l'optimisation combinatoire multi-objectifs et l'optimisation dans l'incertain. En couplant une structure combinatoire, définissant l'ensemble des solutions potentielles, avec un modèle décisionnel pour comparer ces solutions, on obtient un problème d'optimisation avec ses difficultés propres. Dans la section suivante, nous présentons donc différents modèles décisionnels pour la décision multi-objectifs et la décision dans l'incertain, avant d'exposer dans la suite du document les différents algorithmes que nous avons proposés pour optimiser ces modèles sur diverses structures combinatoires.

3 Modèles décisionnels

La théorie de la décision vise à comparer des *alternatives* entre elles, et à proposer et justifier des modèles formels pour décider de la "meilleure" action à entreprendre. Dans les problèmes étudiés en théorie de la décision, on rencontre principalement trois sources de difficulté. Tout d'abord, comme dans l'exemple du classement des universités, il peut y avoir *plusieurs critères* à prendre en considération pour évaluer les différentes actions (critères liés à

la qualité de l'enseignement dispensé, à la production scientifique...). Par ailleurs, les décisions prises peuvent impacter sur *différents acteurs et/ou décideurs* (mise en place d'un nouvel aéroport, d'une nouvelle autoroute) et il s'agit alors souvent de prendre une décision la plus « équitable » possible entre ces différents agents. En outre, les alternatives à comparer peuvent nécessiter la prise en compte explicite de l'*incertitude*, comme lorsque l'on compare des projets de forage pétrolier et que l'on sait mal évaluer la quantité de pétrole disponible. Dans ce dernier cas, on parle de *décision dans l'incertain*.

Face à ces diverses sources de difficulté, nous présentons dans cette section divers modèles décisionnels pour la *décision multi-objectifs* (un objectif représentant un critère ou un agent) et la décision dans l'incertain. Les modèles décisionnels que nous détaillons visent à comparer des vecteurs ou des loteries (vecteurs de récompenses contingentes à des événements), selon que l'on se place en décision multi-objectifs ou en décision dans l'incertain.

3.1 Modèles pour la décision multi-objectifs

Dans un contexte multi-objectifs, chaque alternative $x \in X$ est évaluée dans un espace Y à k dimensions. Par soucis de simplicité, et parce que la plupart de nos travaux s'inscrivent sous cette hypothèse, nous supposons dans la suite que les échelles sur les différentes dimensions sont homogènes (ou normalisées) et à valeurs entières. De plus, dans toute la section, nous supposons que les objectifs sont à maximiser. Nous noterons $f_i(x)$ la valeur de l'alternative x sur la dimension i , et $y = (f_1(x), \dots, f_k(x))$ le vecteur à k dimensions caractérisant la performance d'une alternative x . Comparer des alternatives revient alors à comparer des vecteurs. Afin de comparer des vecteurs, on peut faire appel à une *relation de dominance*, induisant un ordre partiel entre vecteurs, ou à une *fonction d'agrégation*, induisant un ordre complet. Nous présentons dans la suite des relations de dominance et des fonctions d'agrégation couramment rencontrées dans la littérature en décision multicritère et en décision collective.

3.1.1 Décision multicritère

Indépendamment de toute information préférentielle propre au décideur, il est naturel de préférer un vecteur y à un vecteur y' si y est meilleur que y' sur toutes les composantes. Cette relation est appelée *dominance de Pareto*. Un vecteur y domine au sens de Pareto un vecteur y' si :

$$\forall i \in \{1, \dots, k\}, y_i \geq y'_i \text{ et } \exists i \in \{1, \dots, k\}, y_i > y'_i$$

Dans un ensemble X d'alternatives, une alternative x est dite *Pareto-optimale* si il n'existe pas d'alternative x' telle que $f(x')$ domine $f(x)$ au sens de Pareto. Dans un soucis d'efficacité des choix, il est évidemment souhaitable qu'une fonction d'agrégation multicritère φ soit compatible avec la dominance de Pareto, autrement dit que $\varphi(y) > \varphi(y')$ si y domine y' au sens de Pareto. Remarquons que la somme pondérée vérifie cette propriété. Cet opérateur d'agrégation bien connu s'écrit comme suit :

$$s_\lambda(y) = \sum_{i=1}^k \lambda_i y_i$$

où $\lambda = (\lambda_1, \dots, \lambda_k)$ est un vecteur de poids. Cet opérateur souffre toutefois de limites bien connues que nous ne détaillerons pas ici, dont par exemple le fait de ne pas offrir la possibilité de favoriser les profils équilibrés. Une généralisation particulièrement étudiée est l'*intégrale de Choquet* (e.g., Grabisch, 1996), où l'on définit un poids pour tout sous-ensemble de critères. Ce surplus de paramètres préférentiels permet de prendre en compte des interactions positives

ou négatives entre critères (et ainsi favoriser éventuellement les profils équilibrés). Formellement, en notant $N = \{1, \dots, k\}$ l'ensemble des critères, les poids sur les sous-ensembles sont représentés sous la forme d'une *capacité*, c'est-à-dire une fonction d'ensemble $v : 2^N \rightarrow [0, 1]$ telle que $v(\emptyset) = 0$, $v(N) = 1$, et $\forall A, B \in 2^N, A \subseteq B \Rightarrow v(A) \leq v(B)$. Pour tout sous-ensemble $A \subseteq N$, $v(A)$ représente l'importance de la coalition A de critères. Une façon naturelle de faire apparaître la formule de l'intégrale de Choquet en partant de la somme pondérée est de réécrire cette dernière de la façon suivante :

$$s_\lambda(y) = \sum_{i=1}^k \lambda_{(i)} y_{(i)} = \sum_{i=1}^k \lambda(N_{(i)}) (y_{(i)} - y_{(i-1)})$$

où $(\cdot)$ désigne une permutation sur $\{1, \dots, k\}$ telle que $0 = y_{(0)} \leq y_{(1)} \leq \dots \leq y_{(k)}$, $N_{(i)} = \{j \in N : y_j \geq y_{(i)}\} = \{(i), (i+1), \dots, (k)\}$ et $\lambda(N_{(i)}) = \sum_{j \in N_{(i)}} \lambda_j = \sum_{j=i}^k \lambda_{(j)}$. En substituant $\lambda(N_{(i)})$ par $v(N_{(i)})$ (autrement dit en relaxant l'hypothèse d'additivité de la capacité), on obtient précisément la formule de l'intégrale de Choquet :

$$C_v(y) = \sum_{i=1}^k v(N_{(i)}) (y_{(i)} - y_{(i-1)}).$$

Les propriétés vérifiées par une capacité garantissent que l'intégrale de Choquet est bien un opérateur de moyenne (c'est-à-dire qui retourne une valeur comprise entre $y_{(1)}$ et $y_{(k)}$). De plus, il existe toujours un vecteur y Pareto-optimal optimisant $C_v(y)$. Enfin, pour favoriser les profils équilibrés en maximisation, il suffit de prendre une capacité v *supermodulaire*, c'est-à-dire qui vérifie $v(A \cup B) + v(A \cap B) \geq v(A) + v(B)$ pour tout $A, B \subseteq N$ (Chateauneuf et Tallon, 2002).

3.1.2 Décision collective

Une importante littérature en décision collective a pour objet de mesurer les inégalités de revenus au sein d'une population (pour une introduction plus détaillée, voir Moulin, 1988). Plus précisément, il s'agit de comparer des vecteurs de la forme $(y_1, \dots, y_k)$ au sens de l'équité, où y_i représente le revenu de l'agent i . La notion d'équité ne doit pas être confondu avec celle d'égalité, car elle englobe une préoccupation supplémentaire d'efficacité. Un résultat élégant de Chong (1976) (voir aussi Marshall et Olkin, 1979; Shorrocks, 1983), généralisant un résultat plus ancien de Hardy et al. (1934), établit que si un vecteur y domine un vecteur y' au sens de Pareto (y est plus intéressant que y' en termes d'efficacité), ou peut être obtenu à partir de y' en réalisant un transfert de richesse d'un agent i vers un agent j plus pauvre (y est plus intéressant que y' en termes d'égalité), alors y domine y' au sens de la relation de dominance suivante, dite *dominance de Lorenz généralisée* :

$$\forall i \in \{1, \dots, k\}, L_i(y) \geq L_i(y') \text{ et } \exists i \in \{1, \dots, k\}, L_i(y) > L_i(y')$$

où $L_i(y) = \sum_{j=1}^i y_{(j)}$ avec $(\cdot)$ une permutation sur $\{1, \dots, k\}$ telle que $y_{(1)} \leq \dots \leq y_{(k)}$. Ceci correspond bien évidemment à une dominance au sens de Pareto de $L(y)$ sur $L(y')$, où $L(y) = (L_1(y), \dots, L_k(y))$. Inversement, si un vecteur y peut être obtenu à partir d'un vecteur y' en réalisant une séquence d'amélioration(s) au sens de Pareto et/ou de transfert(s) de revenu d'un agent vers un autre agent plus pauvre, alors y domine y' au sens de Lorenz. Par exemple, pour $y = (8, 10, 9, 10)$ et $y' = (5, 10, 11, 8)$, le vecteur y domine y' au sens de Lorenz puisque $L(y) = (8, 18, 27, 37)$ domine $L(y') = (5, 13, 23, 34)$ au sens de Pareto ; on vérifie bien qu'en réalisant un transfert de richesse de 2 du troisième agent vers le premier dans y' , on

obtient $(7, 10, 9, 8)$, puis en enrichissant le premier et dernier agent dans ce vecteur on obtient $(8, 10, 9, 10)$, soit le vecteur y .

Une façon de raffiner la dominance de Lorenz consiste à faire appel aux indices de Gini généralisés (Blackorby et Donaldson, 1978; Gajdos, 2001), qui s'écrivent sous la forme suivante :

$$G_\lambda(y) = \sum_{i=1}^k \lambda_i y_{(i)}$$

avec $\lambda_1 > \lambda_2 > \dots > \lambda_k > 0$. Remarquons que ce critère de décision tend vers le min lorsque l'on pose $\lambda_1 = 1 - \sum_{i=2}^k \varepsilon_i$, $\lambda_2 = \varepsilon_2$, $\dots$, $\lambda_k = \varepsilon_k$ avec $\varepsilon_2 > \dots > \varepsilon_k > 0$ et ε_2 tend vers 0. Un autre cas particulier remarquable est obtenu lorsque l'on pose des poids « à grands pas », c'est-à-dire quand les écarts entre les poids successifs sont très importants ($\lambda_1 \gg \dots \gg \lambda_k$). Le critère de décision se rapproche alors fortement de l'opérateur leximin, qui consiste à comparer deux vecteurs sur la base de leur plus petite composante, puis de leur seconde plus petite composante, et ainsi de suite... Les indices de Gini sont plus connus dans la littérature informatique sous le nom de *moyennes ordonnées pondérées* ou sous l'acronyme OWA, pour *Ordered Weighted Average* (Yager, 1988). Enfin, il est intéressant de noter qu'un OWA est lui-même un cas particulier d'intégrale de Choquet, dans le cas où la capacité est *symétrique*, c'est-à-dire qu'elle vérifie $v(A) = v(B)$ si $|A| = |B|$. Ceci est cohérent dans la mesure où la symétrie reflète que les agents sont traités sur un pied d'égalité. Des travaux récents s'intéressent aux situations où les agents bénéficient de droits exogènes inégaux (par exemple, pour avoir participé en plus ou moins grande proportion au financement d'une ressource commune partagée), mais nous ne les examinerons pas dans le présent document. Le lecteur intéressé pourra se reporter en particulier aux travaux de Bouveret et Lemaître (2007) et d'Ogryczak (2009).

3.2 Modèles pour la décision dans l'incertain

On appelle *décision dans l'incertain* les situations de choix où les conséquences des actions d'un agent ne sont pas connues avec certitude (Savage, 1954). Lorsque cette incertitude est probabilisée, autrement dit lorsque le résultat obtenu ne dépend que de la réalisation d'événements dont on connaît les probabilités, on parle alors de *décision dans le risque* (von Neuman et Morgenstern, 1947; Gayant, 2001). Lorsque l'on ignore la distribution de probabilité sur les événements et que l'on ne peut pas ou ne souhaite pas éliciter des probabilités subjectives, on parle d'*incertain total* ou d'*ignorance complète* (Cohen et Jaffray, 1980). Enfin, dans la situation intermédiaire où la distribution de probabilité est mal connue, c'est souvent le terme d'*ambiguïté* qui est employé (Ellsberg, 1961). C'est le cas par exemple lorsque l'on dispose d'un ensemble de distributions possibles (ensemble défini en extension ou compréhension).

3.2.1 Décision dans le risque

Dans ce cadre, chaque alternative x est caractérisée par une *loterie* (p, y) , c'est-à-dire une distribution de probabilité $(p_1, \dots, p_k)$ sur les conséquences potentielles $(y_1, \dots, y_k)$. De nombreux modèles de décision dans le risque existent dans la littérature. Le plus connu et le plus utilisé est le modèle de l'*espérance d'utilité* (EU), de par sa simplicité et son attrait sur le plan normatif (von Neuman et Morgenstern, 1947). Dans ce modèle, les individus assignent une valeur numérique, nommée utilité, à chaque conséquence, via une fonction d'utilité u . L'évaluation d'une loterie se fait ensuite via un calcul d'espérance d'utilité : étant donnée une loterie $(p, y) = (p_1, y_1; \dots; p_k, y_k)$ (qui conduit à la conséquence y_i avec la probabilité p_i), son espérance d'utilité s'écrit $EU(p, y) = \sum_{i=1}^k p_i u(y_i)$. Une loterie est alors préférée à une autre si son espérance d'utilité est plus grande. Bien que séduisant de par son aspect intuitif, le

modèle EU ne permet pas de rendre compte de tous les comportements décisionnels rationnels. A titre d'illustration, nous présentons ci-dessous une variante du paradoxe d'Allais (1979) due à Kahneman et Tversky (1979).

Exemple 1 (paradoxe d'Allais) *On propose à un agent, dans un premier problème, de choisir entre deux loteries (p, y) et (p', y') , et dans un second problème de choisir entre deux loteries (p'', y'') et (p''', y''') (voir table 1, où les valeurs dans les cases correspondent aux probabilités). Dans le premier problème, l'agent préfère la loterie (p, y) à (p', y') (il préfère*

Loterie	0€	3000€	4000€
y	0.00	1.00	0.00
y'	0.10	0.00	0.90
y''	0.90	0.10	0.00
y'''	0.91	0.00	0.09

TABLE 1 – Loteries dans le paradoxe d'Allais.

être sûr de gagner 3000 € plutôt que risquer de ne rien gagner pour un surplus de gain qui lui semble insuffisant), alors que dans le second problème l'agent préfère la loterie (p''', y''') à la loterie (p'', y'') (car la probabilité de gagner 4000 € avec (p''', y''') est pratiquement la même que la probabilité de gagner 3000 € avec (p'', y'')). Le modèle EU n'est pas capable de rendre compte de ce comportement qu'aurait adopté de nombreux décideurs. En effet, la préférence pour y sur y' implique $u(3000) > 0.1u(0) + 0.9u(4000)$. Ce qui est équivalent à $0.1u(3000) > 0.01u(0) + 0.09u(4000)$, ou encore à $0.9u(0) + 0.1u(3000) > 0.91u(0) + 0.09u(4000)$. Cette dernière inégalité traduit la préférence pour (p'', y'') sur (p''', y''') dans le modèle EU. Ainsi, quelle que soit la fonction d'utilité utilisée, la préférence pour (p, y) sur (p', y') est équivalente à la préférence pour (p'', y'') sur (p''', y''') dans le modèle EU.

Ce type d'exemple contre-intuitif a conduit à l'émergence de modèles non-linéaires plus sophistiqués qui généralisent EU, et en particulier au modèle de l'utilité espérée dépendant du rang (RDU pour *rank dependent utility*), introduit par Quiggin (1982) (voir aussi Quiggin, 1993). Dans le modèle RDU, le poids attribué à une conséquence ne dépend pas uniquement de la probabilité d'obtenir cette conséquence, mais également de l'attractivité du rang de cette conséquence par rapport aux autres conséquences possibles. Ce modèle repose sur deux paramètres : une fonction d'utilité qui est déjà présente dans le modèle EU, et une fonction φ de déformation des probabilités. Il s'agit d'une fonction strictement croissante sur $[0, 1]$ telle que $\varphi(0) = 0$ et $\varphi(1) = 1$. Cette déformation des probabilités ne porte pas sur des probabilités simples, mais sur des cumuls de probabilités. Pour rappel, étant donnée une loterie (p, y) , qui rapporte une utilité $u_i = u(y_i)$ avec une probabilité p_i , on appelle *fonction décumulative* de (p, y) la fonction $G_{(p, y)} : \mathbb{R} \mapsto [0, 1]$ qui associe à chaque utilité α la probabilité d'avoir au moins cette utilité. Plus formellement, $G_{(p, y)}(\alpha) = \sum_{i: u_i \geq \alpha} p_i$. La valeur selon RDU d'une loterie (p, y) est alors définie de la manière suivante :

$$RDU(p, y) = u_{(1)} + \sum_{i=2}^k [u_{(i)} - u_{(i-1)}] \varphi(G_{(p, y)}(u_{(i)}))$$

où $(.)$ correspond à une permutation de $\{1, \dots, k\}$ telle que $u_{(1)} \leq \dots \leq u_{(k)}$. Ce critère peut être interprété comme suit : on est sûr d'obtenir au moins une utilité de $u_{(1)}$, puis on est susceptible d'obtenir un supplément d'utilité de $u_{(2)} - u_{(1)}$ avec une masse de probabilité $\varphi(G_{(p, y)}(u_{(2)}))$, puis un supplément d'utilité de $u_{(3)} - u_{(2)}$ avec une masse de probabilité $\varphi(G_{(p, y)}(u_{(3)}))$, et ainsi de suite... Ce modèle est bien une généralisation du modèle EU. En effet, si la fonction de déformation des probabilités est linéaire, alors on se retrouve dans le cadre du modèle EU. Le modèle RDU permet de résoudre le paradoxe d'Allais (voir exemple 2) et bien d'autres situations où le modèle EU était impuissant.

Exemple 2 Revenons sur le paradoxe d'Allais. On définit pour l'agent une utilité linéaire $u(y) = y$ et on utilise la fonction φ suivante : $\varphi(0.09) = \varphi(0.1) = 0.2, \varphi(0.9) = 0.7$. Les préférences retournées par le modèle RDU sont alors compatibles avec le paradoxe d'Allais :

$$RDU(p, y) = u(3000) = 3000$$

$$RDU(p, y') = u(0) + \varphi(0.9)(u(4000) - u(0)) = 2800$$

Ainsi (p, y) est préférée à (p', y') . De même on a :

$$RDU(p, y'') = u(0) + \varphi(0.1)(u(3000) - u(0)) = 600$$

$$RDU(p, y''') = u(0) + \varphi(0.09)(u(4000) - u(0)) = 800$$

Et donc (p''', y''') est préférée à (p'', y'') .

L'intérêt de déformer des cumuls de probabilités, et non directement les probabilités elles-mêmes (comme c'est par exemple le cas dans le modèle de Handa (1977), qui utilise la formule $\sum_{i=1}^k \varphi(p_i)u(y_i)$), est d'obtenir un critère de choix compatible avec la dominance stochastique. On dit qu'une loterie $(p, y) = (p_1, y_1; \dots; p_k, y_k)$ domine stochastiquement une loterie $(p', y') = (p'_1, y'_1; \dots; p'_k, y'_k)$ si $\forall \alpha \in \mathbb{R}, G_{(p, y)}(\alpha) \geq G_{(p', y')}(\alpha)$, autrement dit, pour tout $\alpha \in \mathbb{R}$, la probabilité d'obtenir une utilité d'au moins α avec la loterie (p, y) est au moins aussi grande qu'avec la loterie (p', y') . La compatibilité avec la dominance stochastique signifie que $RDU(p, y) \geq RDU(p', y')$ dès lors que (p, y) domine stochastiquement (p', y') . Cette propriété est bien entendu souhaitable pour décrire un comportement rationnel, et elle est bien vérifiée par le modèle RDU (contrairement au modèle de Handa).

En outre, l'introduction de la fonction φ de déformation des probabilités offre la possibilité de modéliser finement l'attitude d'un décideur vis-à-vis du risque (pour une description précise des différentes attitudes vis-à-vis du risque, voir Arrow, 1965; Pratt, 1964). En effet, à la différence du modèle EU, le modèle RDU permet de faire la distinction entre être adversaire du risque au sens faible (pour toute loterie $(p, y) = (p_1, y_1; \dots; p_n, y_n)$, la loterie certaine $(1; E(p, y))$ est préférée à (p, y)) et au sens fort (entre deux loteries de même espérance de gain, on préfère celle pour laquelle l'étalement des gains possibles est moindre). Avec le modèle EU, tout adversaire du risque au sens faible l'est aussi au sens fort. Avec le modèle RDU, pour une fonction d'utilité concave, on est adversaire du risque au sens faible si et seulement si $\varphi(p) \leq p$ pour tout $p \in [0, 1]$, et adversaire du risque au sens fort si et seulement si φ est convexe (Quiggin, 1993).

3.2.2 Décision dans l'incertain total

En décision dans l'incertain total, on ignore complètement les probabilités d'occurrence des différentes conséquences. Chaque alternative x est évaluée selon un ensemble S de scénarios possibles, qui peut être défini en extension ou en compréhension. Par exemple, on peut définir en extension un ensemble $S = \{1, 2, 3\}$ de trois scénarios, avec $f_s(x)$ la valeur de l'alternative x dans le scénario s ($s \in \{1, 2, 3\}$). Un autre exemple consiste à définir en compréhension un ensemble $S = \{(\alpha_1, \alpha_2) : \alpha_1 \in [3, 5], \alpha_2 \in [2, 4]\}$, avec $f_s(x) = \alpha_1 + \alpha_2$ la valeur de x dans le scénario $s = (\alpha_1, \alpha_2)$. Les critères classiques de décision dans l'incertain total sont : le critère de Wald (maximin), le critère maximax, le critère de Hurwicz et le critère de Savage. Pour une description détaillée de ces différents critères et de leurs propriétés, le lecteur intéressé pourra se reporter à l'ouvrage de Luce et Raiffa (1957). Un décideur suivant le critère de Wald prendra la décision qui maximise la pire conséquence possible : $W(x) = \min_s f_s(x)$. Inversement, un décideur suivant le critère maximax prendra la décision qui maximise la meilleure conséquence possible : $M(x) = \max_s f_s(x)$. Le premier archétype de décideur peut être qualifié de pessimiste, alors que le second peut être qualifié d'optimiste. Entre ces deux attitudes extrêmes, le critère d'Hurwicz permet de rendre compte d'attitudes plus modérées en s'intéressant à une combinaison linéaire des deux critères précédents : $H(x) =$

$\alpha W(x) + (1 - \alpha)M(x)$. Enfin, le critère de Savage vise à choisir l'alternative qui minimise le regret maximum que l'on pourrait avoir après révélation du scénario qui se réalise. Dans un ensemble X d'alternatives, lorsque le scénario s se réalise, le regret de $x \in X$ se définit par $R(x, s) = \max_{s \in S} (f_s^* - f_s(x))$, où $f_s^* = \max_{x \in X} f_s(x)$. Le regret maximum d'une alternative x s'écrit alors $R(x) = \max_{s \in S} R(x, s)$. L'exemple 3 illustre ces différentes définitions. Il est à noter que les critères de Wald et de Savage ont été largement repris et étudiés, sous les dénominations *minimax* et *minimax regret*, dans toute la littérature sur l'*optimisation robuste* (Kouvelis et Yu, 1997), qui vise à revisiter les problèmes classiques d'optimisation combinatoire en prenant en compte explicitement l'incertitude qu'il peut y avoir sur les paramètres du problèmes, en définissant un ensemble (fini ou non) de scénarios sur les valuations.

Exemple 3 (emprunté à D. Bouyssou) *Considérons un ensemble $X = \{x, x', x'', x'''\}$ d'alternatives, dont les performances dans les différents scénarios sont indiquées dans la matrice de décision de la table 2. Le critère de Wald conduit à choisir x' , le critère maximax ainsi que le critère de Hurwicz pour $\alpha = 0.5$ à choisir x'' , et enfin le critère de Savage à choisir x''' .*

	s_1	s_2	s_3	s_4
x	2	2	0	1
x'	1	1	1	1
x''	0	4	0	0
x'''	1	3	0	0

	s_1	s_2	s_3	s_4
x	0	2	1	0
x'	1	3	0	0
x''	2	0	1	1
x'''	1	1	1	1

TABLE 2 – Matrice de décision (à gauche) et matrice des regrets (à droite).

Pour être rationnel, un critère de décision dans l'incertain total doit être compatible avec la relation de *dominance* suivante : x domine x' si $f(s, x) \geq f(s, x')$ pour tout s et il existe un scénario s' pour lequel $f(s', x) > f(s', x')$. Tous les critères que nous avons présenté sont compatibles avec cette relation dans la mesure où, pour chacun d'entre eux, il existe toujours une alternative non-dominée qui l'optimise.

3.2.3 Décision dans l'incertain avec des probabilités mal connues

Dans de nombreuses situations de choix sous incertitude, on est plus informé que dans l'incertain total, mais on ne dispose toutefois pas d'une distribution de probabilité précise sur les événements pour prendre une décision. Une manière de traiter ce type de situations consiste à prendre en considération un ensemble de distributions de probabilité possibles, plutôt que de parier sur une unique distribution de probabilité. Une décision est alors vue comme un couple (Γ, y) , où $y = (y_1, \dots, y_k)$ désigne l'ensemble des conséquences potentielles (comme en décision dans le risque) et Γ l'ensemble des distributions de probabilité possibles sur ces conséquences. On parle d'*a priori multiples* (*multi-prior* en anglais), selon la terminologie de Gilboa et Schmeidler (1989). Les termes *ambiguïté* (Ellsberg, 1961) et *probabilités imprécises* (Walley, 1991) sont aussi fréquemment employés. Dans la suite, nous appelons *loterie imprécise* un couple (Γ, y) . Pour un panorama complet des travaux en théorie de la décision lorsque les probabilités sont mal connues, on pourra se référer en particulier à l'état de l'art récent de Gilboa et Marinacci (2011). Il est remarquable de noter que, quand bien même on voudrait parier sur une distribution unique pour se ramener au cadre classique de l'utilité espérée, cette dernière n'est parfois pas apte à reproduire les préférences observées, comme l'illustre l'exemple d'Ellsberg (1961).

Exemple 4 (urne d'Ellsberg) Soit une urne comportant $\frac{1}{3}$ de boules rouges et $\frac{2}{3}$ de boules noires ou jaunes. Une boule est tirée dans cette urne et deviner la bonne couleur conduit à un gain de 1 (0 sinon). La plupart des individus vont préférer parier sur rouge plutôt que sur noir, et sur « noir ou jaune » plutôt que sur « rouge ou jaune ». Le lecteur se convaincra sans mal qu'il n'existe pourtant pas de fonction d'utilité u et de distribution de probabilité $p = (p_R, p_N, p_J)$ – avec p_R (resp. p_N, p_J) la probabilité de tirer une boule rouge (resp. noire, jaune) – pour lesquelles $EU(p, y^R) > EU(p, y^N)$ et $EU(p, y^{NJ}) > EU(p, y^{RJ})$, où $(p, y^R) = (p_R, 1; p_N, 0; p_J, 0)$ correspond à la décision de parier sur rouge, et $(p, y^{NJ}) = (p_R, 0; p_N, 1; p_J, 1)$ correspond à la décision de parier sur « noir ou jaune » (les autres notations s'interprétant de façon similaire).

Différents critères de décision peuvent être utilisés pour comparer des décisions sur la base d'un ensemble Γ de distributions de probabilité possibles, qui pour la plupart étendent les critères proposés en décision dans l'incertain total.

Si le décideur cherche à se prémunir contre la pire espérance d'utilité possible, une loterie imprécise (Γ, y) est évaluée selon la formule $\underline{EU}(\Gamma, y) = \min\{EU(p, y) : p \in \Gamma\}$, où $EU(p, y)$ est l'espérance d'utilité de la loterie (p, y) . Ce critère étend le critère de Wald, et il est connu sous le nom de critère Γ -*maximin*. Dans l'exemple d'Ellsberg, l'ensemble Γ est $\{p : p_R = \frac{1}{3}, p_N + p_J = \frac{2}{3}\}$. Supposons pour faciliter les calculs que $u(y_i) = y_i$. La valeur minimum de EU est 0 si l'on parie sur noir (puisque $p_N \in [0, \frac{2}{3}]$), contre $\frac{1}{3}$ si l'on parie sur rouge (puisque $p_R = \frac{1}{3}$). Par ailleurs, la pire espérance d'utilité si l'on parie sur « rouge ou jaune » est $\frac{1}{3}$ (il est possible qu'il n'y ait pas de boule jaune), contre $\frac{2}{3}$ si l'on parie sur « noir ou jaune » (puisque $p_{NJ} = p_N + p_J = \frac{2}{3}$). On reproduit ainsi les préférences observées la plupart du temps.

Inversement, si le décideur cherche à maximiser la meilleure espérance d'utilité possible, une loterie imprécise (Γ, y) est évaluée selon la formule $\overline{EU}(\Gamma, y) = \max\{EU(p, y) : p \in \Gamma\}$. Il s'agit du critère Γ -*maximax*, qui étend le critère maximax. A titre d'illustration, considérons à nouveau l'exemple d'Ellsberg. Selon le critère Γ -maximax, parier sur jaune est préféré à parier sur rouge, car la probabilité p_J vaut au mieux $\frac{2}{3}$, contre $\frac{1}{3}$ pour rouge.

Entre ces deux attitudes extrêmes, à la suite de Jaffray et Jeleva (2007) par exemple, on peut étendre le critère de Hurwicz au cas des probabilités mal connues. Avec le critère de Hurwicz, une loterie imprécise (Γ, y) est évaluée selon la formule $\alpha \underline{EU}(\Gamma, y) + (1 - \alpha) \overline{EU}(\Gamma, y)$. En d'autres termes, le décideur se focalise sur la pire et la meilleure espérance d'utilité possible et, selon son degré α de pessimisme, accorde d'avantage de poids à l'une ou à l'autre. Pour bien distinguer le cas de l'incertain total de celui des probabilités mal connues, nous proposons d'appeler Γ -Hurwicz ce critère.

Notons que, pour chacun de ces critères (Γ -maximin, Γ -maximax et Γ -Hurwicz), il existe toujours une loterie imprécise *non-dominée* qui l'optimise, où la relation de dominance est définie par : (Γ, y) domine (Γ, y') si $EU(p, y) \geq EU(p, y')$ pour toute distribution $p \in \Gamma$ et $EU(p, y) > EU(p, y')$ pour au moins une distribution p . Le sous-ensemble des loteries non-dominées est bien évidemment particulièrement pertinent pour n'importe quel décideur rationnel. Un autre sous-ensemble de loteries particulièrement intéressantes aux yeux d'un décideur est celui des loteries *E-admissibles*. Cette notion a été introduite par Levi (1974) et discutée par Seidenfeld (2004). Une loterie imprécise (Γ, y) est dite *E-admissible* s'il existe au moins une distribution de probabilité $p \in \Gamma$ pour laquelle elle est maximale au sens de l'espérance d'utilité.

3.3 Concilier théorie de la décision et optimisation combinatoire

L'objet de la section suivante sera de montrer comment calculer une (des) solution(s) préférée(s), au sens d'un des modèles décisionnels précédemment décrits, lorsque l'espace des

alternatives a une structure combinatoire. En effet, comme mentionné en section 2, il est fréquent dans les problèmes de décision rencontrés aujourd’hui que l’ensemble des alternatives présente une telle structure. Nous l’illustrons maintenant en donnant un exemple dans chacun des contextes décisionnels décrits au-dessus :

- *décision multicritère* : pour choisir l’itinéraire routier qu’empruntera un convoi exceptionnel de transport de matières dangereuses, on peut vouloir prendre en compte un critère de distance kilométrique, mais aussi un critère de densité de population des zones traversées ;
- *décision collective* : dans le cadre de l’exploitation partagée d’une ressource satellitaire entre plusieurs agents, le choix d’un ensemble de prises de vue compatibles entre elles sur une fenêtre de temps donnée peut se modéliser comme un problème de sac à dos à plusieurs fonctions objectifs (une par agent), chaque agent ne portant pas le même intérêt aux différentes photos ;
- *décision dans l’incertain* : dans un problème d’optimisation de portefeuille boursier, l’ensemble des portefeuilles potentiels est bien évidemment de nature combinatoire, et les conséquences incertaines.

En couplant un modèle décisionnel spécifique avec une structure combinatoire de l’espace des alternatives pour laquelle il n’avait pas été envisagé jusqu’alors, on obtient un nouveau problème. Nos travaux s’inscrivent dans cette logique. Avant d’en donner les grandes lignes dans la suite, nous synthétisons l’essentiel de nos contributions dans le tableau 3, où les modèles décisionnels figurent en ligne et les structures combinatoires en colonnes. À l’intersection de chaque ligne et chaque colonne est indiqué le numéro de la section où la contribution correspondante est évoquée, si elle existe.

		AC	PCC	SD	C	CS	ACS	ADH	DI
MULTI-OBJ.	Pareto	4.1		4.1					
	Lorenz	4.1	4.1						
	OWA	4.2	4.2						
	Choquet	4.2	4.2	4.2					
INCERTAIN	EU					5	5		
	RDU		5.1					5.2	5.2
	Savage	5.1	5.1		5.1				
	Γ -Hurwicz							5.2	

TABLE 3 – Organisation de la suite du document. AC : Arbre Couvrant, PCC : Plus Court Chemin, SD : Sac à Dos, C : Couplage, CS : Couplage Stochastique, ACS : Arbre Couvrant Stochastique, ADH : Arbre Décision Hasard, DI : Diagramme d’Influence.

4 Aspects computationnels de la décision multi-objectifs

Le mariage de la décision multi-objectifs et de l’optimisation combinatoire a donné lieu à des développements nombreux, sous l’étiquette d’*optimisation combinatoire multi-objectifs*, avant même l’émergence de la théorie de la décision algorithmique. Comme son nom l’indique, l’objet de l’optimisation combinatoire multi-objectifs est de traiter de problèmes d’optimisation combinatoire en présence de points de vue multiples (pour une synthèse voir Ehrgott, 2005). Plus formellement, la structure de valuation de tels problèmes est constituée de vecteurs, en lieu et place des valuations scalaires de l’optimisation mono-objectif standard. Tous les problèmes classiques de l’optimisation combinatoire (problèmes de graphes, programmation

linéaire en nombres entiers...) peuvent être revisités dans ce cadre, et de nouveaux algorithmes doivent être proposés. Deux types d'approches peuvent être étudiées : soit on cherche à générer l'ensemble des solutions Pareto-optimales (approche « historique » en optimisation multi-objectifs), soit on cherche à déterminer une solution de compromis optimisant une fonction d'agrégation des critères (approche qui s'est vue confortée par le développement de la théorie de la décision algorithmique). Chercher à calculer le front de Pareto est naturel lorsque l'on ne dispose pas d'information préférentielle ou lorsque l'information préférentielle dont on dispose ne permet pas d'élucider les paramètres d'une fonction d'agrégation : les solutions Pareto-optimales sont en effet les seules susceptibles d'être choisies par un décideur rationnel. L'intérêt pour cette approche a donné lieu à une importante littérature (pour un état de l'art sur le sujet, voir Ehrgott et Gandibleux, 2000, 2004). Néanmoins, le nombre de solutions Pareto-optimales peut augmenter exponentiellement avec la taille de l'instance (e.g. Hamacher et Ruhe, 1994; Hansen, 1980) et le nombre d'objectifs (Rosinger, 1991). En conséquence, pour des instances de tailles réelles, cette approche peut devenir rapidement impraticable quand le nombre d'objectifs dépasse trois. Cela justifie l'effort que nous avons porté aux développements d'algorithmes d'optimisation pour des modèles décisionnels moins frustrés que la dominance de Pareto. Dans cette section, nous décrivons tout d'abord nos travaux pour déterminer les solutions Pareto-optimales dans des problèmes bi-objectifs (section 4.1), avant de détailler nos contributions dans le cadre de la recherche d'une unique solution de compromis (section 4.2).



Avertissement de lecture. Afin d'éviter que l'auteur du présent mémoire ne sombre dans la trichotillomanie, certains résultats de cette section sont présentés en minimisation, et d'autres en maximisation. Pour faciliter la lecture, nous prenons toutefois bien soin en début de chaque partie d'indiquer le choix de présentation qui a été fait.

4.1 Détermination de la frontière de Pareto

L'approche la plus répandue en optimisation combinatoire multi-objectifs vise à générer le front de Pareto dans son entier. La plupart des méthodes classiques exactes ou approchées pour les problèmes d'optimisation combinatoire mono-objectif ont été revisitées afin de proposer une version multi-objectifs, que ce soient la programmation dynamique (Daellenbach et De Kluyver, 1980; Klamroth et Wiecek, 2000), les méthodes de *branch and bound* (Bitran et Rivera, 1982; Kiziltan et Yucaoglu, 1983; Marcotte et Soland, 1986; Mavrotas et Diakoulaki, 1998) ou encore les algorithmes gloutons (Serafini, 1986), ainsi que de nombreuses approches heuristiques et métaheuristiques (Ehrgott et Gandibleux, 2004).

Comme souligné par Ehrgott et Gandibleux (2000), parmi les méthodes génériques de résolution, le *branch and bound* a fait finalement l'objet de peu de tentatives de généralisation au cas multi-objectifs au regard de son utilisation intensive en optimisation combinatoire mono-objectif. Ce type de méthode allie pourtant l'optimalité des solutions retournées (comme la programmation dynamique ou les algorithmes gloutons lorsque le problème étudié remplit les conditions requises pour leur validité) à l'adaptabilité à un large ensemble de problèmes (comme les métaheuristiques). Ceci nous a poussé à nous intéresser à cette méthode de résolution pour en proposer une extension opérationnelle au cas multi-objectifs. Nous présentons l'extension que nous avons proposée dans la section 4.1.1.

Par ailleurs, notons que toutes les méthodes de résolution proposées pour déterminer la frontière de Pareto seront au mieux de complexité pseudo-polynomiale dans le cas général, de par la taille de la frontière de Pareto qui est elle-même généralement pseudo-polynomiale (i.e., elle est bornée par une fonction polynomiale de la plus grande valeur présente dans l'instance). En effet, si la valeur sur un objectif d'une solution réalisable est bornée par M , il y a M^p

points entiers distincts dans l'espace des objectifs et donc jusqu'à M^{p-1} points non-dominés. Néanmoins, il existe toujours une frontière ε -approchée (cette notion sera définie dans la suite) de cardinal polynomial en $1/\varepsilon$ et en la taille de l'instance (Papadimitriou et Yannakakis, 2000). Hormis ce résultat d'existence, Papadimitriou et Yannakakis (2000) ont également présenté un schéma complet d'approximation polynomiale (FPTAS en anglais) pour calculer une telle frontière ε -approchée dans divers problèmes multi-objectifs (plus court chemin, arbre couvrant minimum, couplage). Néanmoins, comme souligné par Bazgan et al. (2007b), les concepteurs de FPTAS, généralement issus de l'informatique théorique, se focalisent souvent plus sur l'existence d'un schéma que sur sa mise en œuvre opérationnelle. Une voie de recherche originale consiste donc à développer et implémenter des FPTAS efficaces en pratique pour l'optimisation multi-objectifs. Notre contribution sur ce sujet fait l'objet de la section 4.1.2.

Une autre voie pour réduire la taille de l'ensemble retourné consiste à s'intéresser à un sous-ensemble de la frontière de Pareto. Ainsi, dans le cas où les objectifs représentent des agents, on peut vouloir focaliser la recherche sur la *frontière de Lorenz*. La frontière de Lorenz désigne l'ensemble des solutions non-dominées au sens de la dominance de Lorenz généralisée telle que définie en section 3.1.2. Rappelons que la frontière de Lorenz prend tout son sens en décision collective, et qu'elle est un sous-ensemble de la frontière de Pareto. Ce sous-ensemble comporte les solutions « équitables » en un certain sens.

Dans la suite, nous notons $\mathcal{X}$ l'ensemble des solutions réalisables, et $\mathcal{Y}$ son image dans l'espace des objectifs. L'image d'une solution $x \in \mathcal{X}$ est $f(x) = (f_1(x), \dots, f_k(x))$. L'ensemble des solutions Pareto-optimales est noté $\mathcal{X}^*$, et son image dans l'espace des objectifs est notée $\mathcal{Y}^* = \{f(x) : x \in \mathcal{X}^*\}$. Les solutions non-dominées qui optimisent une somme pondérée des objectifs sont dite *supportées*. Une solution supportée est dite *extrême* si son image est un sommet non-dominé de l'enveloppe convexe de $\mathcal{Y}$. Lorsque l'on parle de déterminer la frontière de Pareto, on ne vise pas à calculer *toutes* les solutions Pareto-optimales, mais plutôt un ensemble *complet* de solutions (Hansen, 1980), c'est-à-dire un ensemble comportant *une seule* solution³ par vecteur non-dominé dans $\mathcal{Y} = f(\mathcal{X}) = \{f(x) : x \in \mathcal{X}\}$.

Exemple 5 *Considérons le problème de sac-à-dos bi-objectifs suivant :*

$$\begin{aligned} \max \quad & \begin{cases} 10x_1 + 2x_2 + 6x_3 + 9x_4 + 12x_5 + x_6 \\ 2x_1 + 7x_2 + 6x_3 + 4x_4 + x_5 + 3x_6 \end{cases} \\ \text{s.c.} \quad & 4x_1 + 4x_2 + 5x_3 + 4x_4 + 3x_5 + 2x_6 \leq 6 \\ & x_j \in \{0, 1\} \quad j \in \{1, \dots, 6\} \end{aligned}$$

Les solutions Pareto-optimales sont :
 $\mathcal{X}^* = \{(0, 0, 0, 0, 1, 1), (1, 0, 0, 0, 0, 1), (0, 0, 0, 1, 0, 1), (0, 1, 0, 0, 0, 1)\}$, *et leur image dans l'espace des objectifs est*
 $\mathcal{Y}^* = \{(13, 4), (11, 5), (10, 7), (3, 10)\}$ *(voir la figure 3). Remarquons que toutes les solutions de $\mathcal{X}^*$ ont des images distinctes dans l'espace des objectifs, et donc $\mathcal{X}^*$ est un ensemble complet de solutions.*

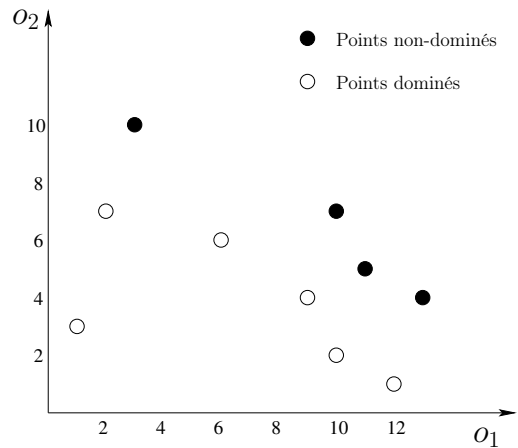


FIGURE 3 – Espace des objectifs.

3. Remarquons que de nombreux travaux dans la littérature visent à déterminer *toutes* les solutions Pareto-optimales. A notre sens, ce but ne coïncide pas avec celui poursuivi en optimisation combinatoire mono-objectif, où l'on cherche à déterminer *une* solution optimale, et non toutes les solutions optimales.

4.1.1 Branch and bound multi-objectifs

Les résultats sur les bornes multi-objectifs sont présentés en maximisation (on procède symétriquement en minimisation), et les résultats sur la mise en œuvre de ces bornes sont présentés en maximisation ou minimisation selon la nature des problèmes étudiés.

Afin de développer un algorithme de *branch and bound* multi-objectifs, il est nécessaire d'introduire des notions de bornes supérieures et inférieures pour l'optimisation multi-objectifs. Cela a été fait plusieurs fois dans la littérature. A la différence du cas mono-objectif où une borne est définie par une unique valeur scalaire, une borne multi-objectifs est un ensemble de vecteurs. En effet, de par la nature partielle de l'ordre induit par la dominance de Pareto, faire appel à un ensemble de vecteurs plutôt qu'à un unique vecteur pour définir une borne permet d'approcher beaucoup plus finement l'image des solutions potentielles dans l'espace des objectifs. À la suite de Ehrgott et Gandibleux (2007), nous avons proposé de calculer des bornes multi-objectifs en se fondant sur la partie non-dominée de l'enveloppe convexe de l'image des solutions dans l'espace des objectifs (Sourd et Spanjaard, 2008). Le principe en est décrit ci-dessous, avec illustration sur le problème du sac-à-dos bi-objectifs (Delort et Spanjaard, 2010).

Borne supérieure multi-objectifs. La première idée qui vient à l'esprit pour borner supérieurement un ensemble $Y \subseteq \mathcal{Y}$ de vecteurs (représentant par exemple l'image d'un sous-ensemble de solutions dans un *branch and bound* multi-objectifs) est de définir un unique vecteur y^I tel que $y_i^I = \max_{y \in Y} y_i$ pour $i = 1, \dots, k$. Ce point est appelé le *point idéal* de Y , dont les coordonnées sont donc les valeurs optimales sur chacune des composantes prises séparément. Néanmoins, ce point idéal est souvent très « loin » des points de Y . Pour cette raison, il est utile de définir une borne supérieure à partir d'un *ensemble* de vecteurs plutôt qu'à partir d'un seul élément. À la suite de Villareal et Karwan (1981), nous disons qu'un ensemble $\mathbf{UB}$ est une borne supérieure multi-objectifs de Y si :

$$\forall y \in Y, \exists y' \in \mathbf{UB}, y'_i \geq y_i \text{ pour tout } i = 1, \dots, k.$$

Remarquons que cette définition est compatible avec le cas mono-objectif (une borne étant alors définie par un singleton), et qu'une définition identique (en version minimisation) a été proposée en intelligence artificielle dans le cadre du développement d'un algorithme de recherche heuristique multi-objectifs (Stewart et White III, 1991). Cependant, dans ces travaux et les suivants, aucune procédure opérationnelle n'a été proposée pour calculer des bornes multi-objectifs qui ne se réduisent pas à un unique vecteur (plus précisément, le point idéal). A la suite de Ehrgott et Gandibleux (2007), nous avons proposé de calculer des bornes supérieure multi-objectifs en définissant $\mathbf{UB}$ comme la frontière non-dominée de l'enveloppe convexe de Y . Cette frontière peut être caractérisée de manière compacte en stockant uniquement les sommets non-dominés de l'enveloppe convexe. Un algorithme récursif proposé par Aneja et Nair (1979) permet de calculer ces sommets efficacement dans le cas bi-objectifs, lorsqu'il existe un algorithme de résolution polynomial ou pseudo-polynomial pour le cas mono-objectif. Cette méthode procède en effet en résolvant une version mono-objectif du problème pour déterminer chaque point extrême. Le nombre de fois que l'algorithme de résolution mono-objectif est lancé est par conséquent linéaire du nombre de points extrêmes.

Exemple 6 Dans l'instance de sac-à-dos bi-objectifs de l'exemple 5, supposons que l'on cherche à borner supérieurement l'ensemble $X_{\bar{6}} \subseteq \mathcal{X}$ des solutions réalisables où l'objet 6 n'a pas été sélectionné (plus précisément, borner son image $f(X_{\bar{6}})$ dans l'espace des objectifs). La méthode d'Aneja et Nair retourne l'ensemble $\{(12, 1), (9, 4), (6, 6), (2, 7)\}$ comme points extrêmes

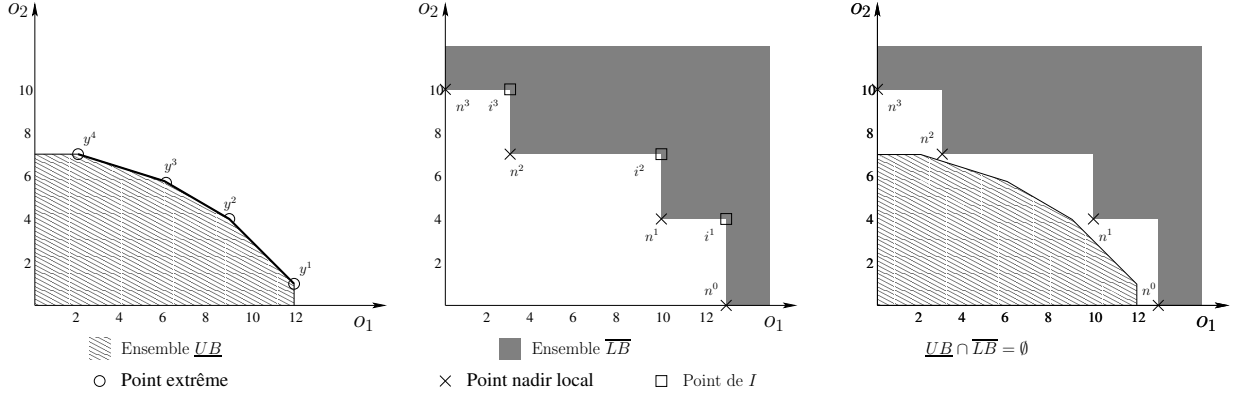


FIGURE 4 – Bornes multi-objectifs dans le cas bi-objectifs.

de $Y_{\bar{6}} = f(X_{\bar{6}})$, soit les points y^1, y^2, y^3, y^4 sur le graphique de gauche de la figure 4. La borne supérieure correspondante est la ligne brisée représentée en gras sur le même graphique.

Borne inférieure multi-objectifs. De façon similaire à une borne supérieure multi-objectifs, la première idée qui vient à l'esprit pour borner inférieurement un ensemble Y de vecteurs est de définir un unique vecteur y^A tel que $y_i^A = \min_{y \in Y} y_i$ pour $i = 1, \dots, k$. Ce point est appelé le *point anti-idéal* de Y . Ici encore, prendre plusieurs points en compte simultanément permet de borner plus finement l'ensemble Y . Nous disons qu'un ensemble $\mathbf{LB}$ est une borne inférieure multi-objectifs de Y si :

$$\forall y \in Y, \exists y' \in \mathbf{LB}, y'_i \leq y_i \text{ pour tout } i = 1, \dots, k.$$

Comme au-dessus, cette définition est compatible avec le cas mono-objectif. Dans un branch and bound multi-objectifs, supposons que l'on souhaite borner inférieurement l'ensemble $f(\mathcal{X}^*) = \mathcal{Y}^*$ sur la base de l'image $I \subseteq \mathcal{Y}$ des solutions trouvées jusqu'alors.

Afin d'obtenir d'obtenir une borne inférieure multi-objectifs de $\mathcal{Y}^*$ ne se limitant pas à un singleton, il est possible de procéder comme suit dans le cas bi-objectifs : si $\{(i_1^j, i_2^j) : 1 \leq j \leq p\}$ constitue les points de I^* (points non-dominés de I) triés dans l'ordre lexicographique (c'est-à-dire décroissant du premier critère, et croissant du second), une borne inférieure multi-objectifs de $\mathcal{Y}^*$ peut être définie en posant $\mathbf{LB} = \mathcal{N}(I) = \{n^j = (i_1^{(j+1)}, i_2^j) : 0 \leq j \leq p\}$, où $i_2^0 = 0$ et $i_1^{(p+1)} = 0$ (voir les croix sur le graphique du milieu de la figure 4). L'ensemble $\mathcal{N}(I)$ peut ici être vu comme une généralisation du *point nadir* de I (dont les composantes sont les pires valeurs possibles parmi les points de I^*). Les points de $\mathcal{N}(I)$ sont à ce titre parfois appelés *points nadirs locaux* (Ehrgott et Gandibleux, 2007).

Exemple 7 Revenons encore une fois sur l'exemple 5, et considérons le sous-ensemble suivant de points de $\mathcal{Y}$: $I = \{(13, 4), (10, 7), (3, 10)\}$. La borne inférieure multi-objectifs de $\mathcal{Y}^*$ est alors : $\mathcal{N}(I) = \{(13, 0), (10, 4), (3, 7), (0, 10)\}$, soit les points $\{n^0, n^1, n^2, n^3\}$ sur le graphique du milieu de la figure 4.

Comparer des bornes multi-objectifs. Les procédures de *branch and bound* visent à écarter des sous-ensembles entiers de solutions en faisant appel à des règles simples. Cela nécessite de pouvoir déterminer efficacement si un sous-ensemble $X \subseteq \mathcal{X}$ de solutions réalisables est susceptible de comporter des solutions Pareto-optimales dans $\mathcal{X}$. Pour ce faire, on compare

une borne supérieure $\mathbf{UB}$ de $f(X)$ et une borne inférieure $\mathbf{LB}$ de $f(\mathcal{X}^*) = \mathcal{Y}^*$. À la différence du cas mono-objectif, cette comparaison n'est pas triviale dans le cas multi-objectifs car on manipule des ensembles en lieu et place de scalaires. Les deux définitions suivantes permettent de réaliser cette opération simplement dans un cadre multi-objectifs : étant donnée une borne supérieure $\mathbf{UB}$, l'ensemble $\underline{\mathbf{UB}}$ est défini par $\underline{\mathbf{UB}} = \{y \in \mathbb{R}_+^m, \exists u \in \mathbf{UB}, u_i \geq y_i \forall i\}$, et l'ensemble $\overline{\mathbf{LB}}$ est défini par $\overline{\mathbf{LB}} = \{y \in \mathbb{R}_+^m, \exists l \in \mathbf{LB}, y_i \geq l_i \forall i\}$. Il est clair que $\underline{\mathbf{UB}} \supseteq f(X)$ et $\overline{\mathbf{LB}} \supseteq \mathcal{Y}^*$. Par conséquent $\underline{\mathbf{UB}} \cap \overline{\mathbf{LB}} = \emptyset$ implique que $f(X) \cap \mathcal{Y}^* = \emptyset$. Dans ce cas, le sous-ensemble X de solutions peut bien sûr être écarté.

Exemple 8 *Revenons une dernière fois sur l'exemple 5. Supposons que les points $I = \{(13, 4), (10, 7), (3, 10)\} \subseteq \mathcal{Y}$ sont déjà connus. Supposons que l'on cherche à savoir si l'ensemble $Y_{\bar{6}}$ est susceptible de comporter des points qui ne soient pas dominés (au sens large) par un point de I . L'ensemble $\underline{\mathbf{UB}}$ est représenté sur le graphique de gauche de la figure 4. L'ensemble $\overline{\mathbf{LB}}$ (déterminé à partir de I) est représenté sur le graphique du milieu. On constate sur le graphique de droite que l'intersection des deux ensembles est vide. L'ensemble $X_{\bar{6}}$ ne peut donc pas comporter de nouvelles solutions Pareto-optimales, et n'a pas besoin d'être exploré.*

Mises en œuvre. Ces bornes multi-objectifs nous ont permis de développer, d'une part, un *branch and bound* très performant pour le problème de l'arbre couvrant minimum bi-objectifs (Sourd et Spanjaard, 2008), d'autre part une nouvelle approche pour le problème du sac-à-dos bi-objectifs par hybridation de la programmation dynamique et du *branch and bound* multi-objectifs (Delort et Spanjaard, 2010). Des travaux sont en cours sur le problème d'affectation bi-objectifs (Delort et Spanjaard, 2011), que nous ne détaillerons pas ici.

Arbre couvrant bi-objectifs. Le problème de l'arbre couvrant minimum multi-objectifs (pour un état de l'art détaillé, voir Ruzika et Hamacher, 2009) a été introduit par Corley (1985), et prouvé NP-difficile par Emelichev et Perepelitsa (1988) (voir aussi Hamacher et Ruhe, 1994), même pour deux objectifs seulement. Plusieurs méthodes d'approximation de la frontière de Pareto (sans garantie de performance) ont été proposées (Hamacher et Ruhe, 1994; Zhou et Gen, 1999; Knowles et Corne, 2001). Concernant les méthodes exactes, à notre connaissance, seules deux méthodes avaient été proposées (Ramos et al., 1998; Steiner et Radzik, 2008), toutes deux spécifiques au cas bi-objectifs et fondées sur une procédure (exacte) en deux phases (Visée et al., 1998). Dans ces méthodes, on calcule d'abord les solutions Pareto-optimales extrêmes dans une première phase (les sommets de l'enveloppe convexe des solutions dans l'espace des objectifs), puis on calcule l'ensemble des solutions Pareto-optimales non-extrêmes dans une deuxième phase, en se restreignant aux seules zones de l'espace des objectifs susceptibles de comporter de nouveaux points non-dominés, appelées zones de recherche dans la suite. Ceci est illustré sur la figure 5 (rappelons que le sens d'optimisation est ici la minimisation). Les méthodes diffèrent principalement dans la façon dont sont calculées les solutions Pareto-optimales non-extrêmes : dans la méthode de Ramos et al. (1998), ces solutions sont déterminées en lançant un *branch and bound* pour chaque zone de recherche ; dans la méthode de Steiner et Radzik (2008), ces solutions sont déterminées en lançant un algorithme d'énumération ordonnée des solutions (appelé parfois aussi algorithme des k -meilleurs) pour chacune de ces zones. Les deux méthodes ont été implémentées par Steiner et Radzik : la dernière s'est avérée significativement plus rapide après avoir été testée sur des cliques (graphes complets) comportant jusqu'à 38 sommets (résolution en une minute environ en moyenne), et sur des graphes en grille ou planaires comportant jusqu'à 121 sommets (résolution en moins de dix minutes en moyenne).

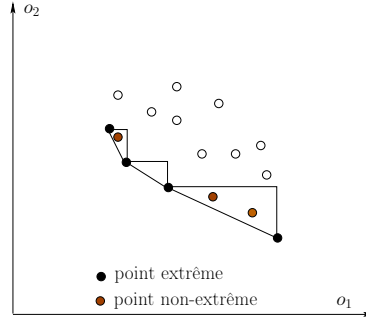


FIGURE 5 – Les triangles sont les seules zones de l’espace des objectifs susceptibles de comporter de nouveaux points non-dominés.

Nous avons développé un *branch and bound* multi-objectifs pour ce problème (Sourd et Spanjaard, 2008). Dans notre approche, l’ensemble **UB** est initialisé à l’aide d’une procédure approchée en deux phases, similaire à celle développée par Andersen et al. (1996) : les solutions extrêmes sont calculées dans un premier temps, puis une recherche locale est lancée à partir de chacune des solutions extrêmes. Pour l’immense majorité des instances aléatoires générées (cliques avec arêtes bivaluées dans $\{1, \dots, 100\}$), la frontière exacte est retournée à l’issue de cette procédure. L’objet du *branch and bound* multi-objectifs est donc plus de certifier l’exactitude de la frontière que de la calculer. Le principe de branchement est très simple : en chaque nœud du *branch and bound*, une arête e est sélectionnée (selon une heuristique que nous ne détaillerons pas ici) et on crée deux sous-problèmes. Dans le premier sous-problème, l’arête e est rendue obligatoire, alors que dans le second elle est rendue interdite. Concernant le principe d’évaluation des sous-ensembles de solutions ainsi caractérisés, le calcul des points extrêmes (pour la borne) est grandement facilité par le fait, d’une part, que l’arbre couvrant minimum mono-objectif est un problème polynomial, et d’autre part, que le nombre de points extrêmes est lui-même polynomial dans les problèmes d’arbres couvrants bi-objectifs.

L’explication de cette dernière propriété est assez simple une fois trouvée la bonne représentation graphique. Prenons un exemple. Considérons un graphe à $m = 7$ arêtes valuées $(\alpha_1, \beta_1), \dots, (\alpha_7, \beta_7)$. Associons à chaque arête i un segment reliant les points de coordonnées α_i et β_i sur deux axes parallèles, comme représenté sur la figure 6. Pour obtenir le rangement des arêtes pour la somme pondérée $\lambda\alpha_i + (1 - \lambda)\beta_i$, il suffit de tracer la droite de niveau λ comme indiqué sur le graphique : le rangement des arêtes correspond à l’ordre dans lequel les segments sont intersectés par la droite (pour $\lambda = 1$ (resp. $\lambda = 0$), on obtient le rangement des arêtes selon les α_i (resp. β_i)). Le nombre de rangements est donc borné par le nombre de points d’intersection des m segments $[\alpha_i, \beta_i]$, c’est-à-dire qu’il est en $O(m^2)$.

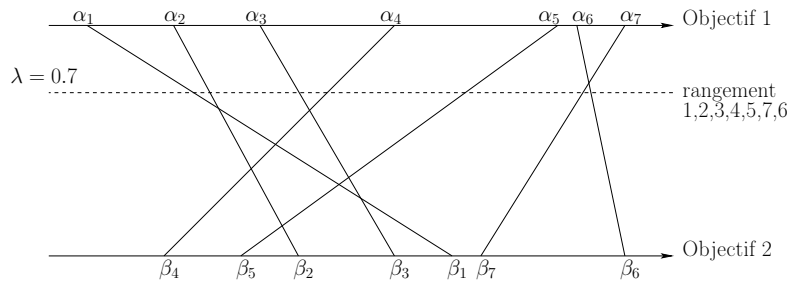


FIGURE 6 – Le nombre de tris des arêtes est polynomial.

Les tests numériques menés ont montré qu’avec ce *branch and bound* on pouvait déterminer le front de Pareto en moins d’une minute en moyenne sur des instances comportant jusqu’à 200 sommets, et en moins de dix minutes en moyenne sur des instances comportant jusqu’à 400 sommets, aussi bien pour les grilles que pour les cliques. Pour les instances de type clique, la quasi-intégralité du temps de résolution est consacré à la recherche locale initiale, le temps dédié au *branch and bound* proprement dit étant négligeable comparativement (jamais plus de dix secondes pour toutes les tailles d’instances). Pour les instances de type grille, le temps de résolution est réparti équitablement entre l’initialisation et le *branch and bound*.

Sac-à-dos bi-objectifs. Visée et al. (1998) ont proposé une méthode en deux phases pour résoudre le problème du sac-à-dos bi-objectifs. Les solutions non-extrêmes sont calculées en lançant un *branch and bound* par zone de recherche susceptible de comporter des solutions Pareto-optimales non-extrêmes. Depuis ce travail, d’autres approches exactes plus performantes ont été proposées : une approche par étiquetage qui procède par réduction du problème à un plus court chemin multi-objectifs (Captivo et al., 2003), et une approche par programmation dynamique proposée par Bazgan et al. (2007a, 2009).

Notre contribution sur ce problème consiste en un algorithme de programmation dynamique multi-objectifs couplé à des règles de coupe faisant appel aux bornes multi-objectifs (Delort et Spanjaard, 2010). La programmation dynamique multi-objectifs s’appuie sur le principe d’optimalité suivant : toute sous-politique d’une politique Pareto-optimale est Pareto-optimale. Dans une procédure de programmation dynamique multi-objectifs, on calcule donc, de façon récursive, des ensembles de vecteurs. L’intégration de règles de coupes durant la programmation dynamique permet de réduire le nombre d’éléments manipulés. L’hybridation que nous proposons est dans l’esprit des règles de dominance entre états employées dans les travaux de Bazgan et al. (2007b, 2009), mais permet de réaliser des économies très importantes en espace mémoire, et d’améliorer significativement les temps de résolution. Une version en deux phases de notre algorithme fournit des résultats encore meilleurs grâce à une procédure de *shaving* (Martin et Shmoys, 1996), où, préalablement à l’exploration d’une zone de recherche par programmation dynamique, on prétraite l’instance en testant chaque objet à l’aide des bornes multi-objectifs. Le principe du *shaving* est simple : avant même le début de la procédure de résolution proprement dite, qui a pour objectif d’énumérer les solutions Pareto-optimales dans une zone de recherche donnée, on rend obligatoire (resp. interdit) un objet, et si l’on détecte que le sous-ensemble de solutions ainsi caractérisé n’est pas susceptible de comporter de solution Pareto-optimale dans la zone de recherche visée, alors l’objet est fixé comme interdit (resp. obligatoire). En répétant l’opération pour chaque objet, cela permet souvent une réduction conséquente de la taille de l’instance.

4.1.2 Approximation avec garantie de performance

Nous présentons les résultats de cette section en minimisation.

Tout comme la notion de borne, la notion d’approximation a été étendue au cas multi-objectifs. Encore une fois, la difficulté supplémentaire vient de ce qu’il s’agit d’approximer des *ensembles* de vecteurs, et non plus une performance scalaire. Pour ce faire, on s’appuie sur les concepts de *couverture* et d’ ε -dominance. On dit qu’un ensemble Y' de points couvre un ensemble Y au sens d’une relation de dominance R si pour tout $y \in Y$ il existe $y' \in Y'$ tel que $y'Ry$. Par ailleurs, on dit qu’un vecteur y' ε -domine un vecteur y si $y'_i \leq (1 + \varepsilon)y_i \forall i$. Dans un problème d’optimisation combinatoire multi-objectifs, un algorithme d’approximation avec garantie de performance vise à retourner un ensemble X_ε de solutions tel que $Y_\varepsilon = f(X_\varepsilon)$ couvre $\mathcal{Y}^*$ au sens de la relation d’ ε -dominance.

A titre d'illustration, considérons la partie gauche de la figure 7, concernant un problème bi-objectifs où chaque solution réalisable est représentée par un point $y = (y_1, y_2)$ dans l'espace des objectifs. Au sein de cet espace, le point p^1 (resp. p^2) ε -domine tous les points dans le cône C^1 (resp. C^2). La notion d' ε -couverture arrive alors naturellement. En effet, l'ensemble $\{p^1, p^2\}$ est une ε -couverture à 2 points du nuage de points Y puisque $Y \subseteq C^1 \cup C^2$. Remarquons qu'un ε plus petit conduit à une ε -couverture plus fine, comme illustré sur la partie droite de la figure 7, où une ε -couverture à 5 points du même ensemble Y est donnée.

Notons que, pour un ε fixé, une ε -couverture d'un ensemble n'est généralement pas unique. Par exemple, considérons un ensemble $Y = \{y, y', y''\}$ avec $y = (800, 950)$, $y' = (880, 880)$ et $y'' = (950, 800)$ et posons $\varepsilon = 0.1$. L'ensemble $\{y, y''\}$ est une ε -couverture de Y puisque y ε -domine y' ($800 \leq 968 = (1 + 0.1) \times 880$ et $950 \leq 968$). Précisons que l'on n'a ni y ε -domine y'' , ni l'inverse, et donc que ni $\{y\}$ ni $\{y''\}$ ne constituent une ε -couverture de Y à eux seuls. Par contre, le singleton $\{y'\}$ constitue lui une autre ε -couverture de Y . En effet, on vérifie d'une part que y' ε -domine y ($880 \leq 880 = (1 + 0.1) \times 800$ et $880 \leq 1045 = (1 + 0.1) \times 950$), d'autre part que y' ε -domine y'' (pour les mêmes raisons).

On est toutefois assuré qu'il existe toujours une ε -couverture de taille raisonnable, grâce à la propriété suivante : pour tout nombre $k > 1$ fixé d'objectifs, pour tout $\varepsilon > 0$ fini et pour tout ensemble $\mathcal{Y}$ de vecteurs y tels que $1 \leq y_i \leq M \forall i$, il existe une couverture de $\mathcal{Y}^*$, au sens de l' ε -dominance, dont la taille est polynomiale en $\log M$ et $1/(\log(1 + \varepsilon))$ (voir Papadimitriou et Yannakakis, 2000; Laumanns et al., 2002). Cela se prouve facilement en considérant une fonction d'échelle logarithmique $\varphi : \mathbb{Z}_+^k \rightarrow \mathbb{Z}_+^k$ sur l'espace des objectifs, définie par :

$$\varphi(y) = (\varphi_1(y), \dots, \varphi_k(y)) \text{ avec } \varphi_i(y) = \left\lfloor \frac{\log y_i}{\log(1 + \varepsilon)} \right\rfloor$$

Pour chaque composante y_i , la fonction φ retourne un entier p tel que $(1 + \varepsilon)^p \leq y_i < (1 + \varepsilon)^{p+1}$. L'image selon φ d'un vecteur y correspond donc aux coordonnées de y à travers une « grille logarithmique », comme représentée sur la figure 8. Etant donnés deux vecteurs y, y' , remarquons que si $\varphi(y) = \varphi(y')$, autrement dit si y et y' se situent dans la même case de la grille logarithmique, alors y ε -domine y' et inversement. Par conséquent, en sélectionnant un point par case non-vide de la grille, on obtient une couverture de $\mathcal{Y}$, et donc de $\mathcal{Y}^*$. La partie gauche de la figure 8 illustre ce point sur l'exemple bi-objectifs introduit en figure 7. La taille de cette ε -couverture est bornée supérieurement par le nombre de cases de la grille, qui n'est pas plus grand que $(1 + \lfloor \log M / \log(1 + \varepsilon) \rfloor)^k$ (Papadimitriou et Yannakakis, 2000). Une ε -couverture de cardinal moindre peut être obtenue facilement en supprimant les « cases dominées », comme illustré sur la partie droite de la figure 8, ce qui améliore la borne à $(1 + \lfloor \log M / \log(1 + \varepsilon) \rfloor)^{k-1}$ (voir Laumanns et al., 2002).

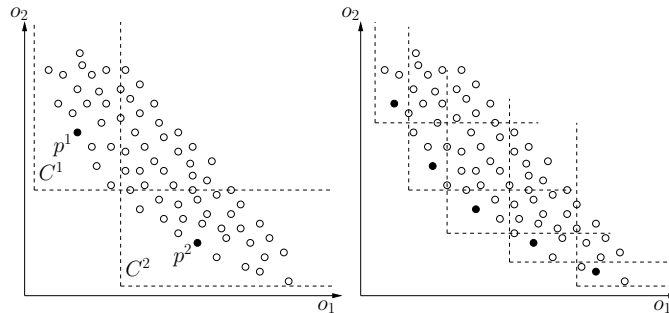


FIGURE 7 – ε -couvertures pour deux valeurs de ε .

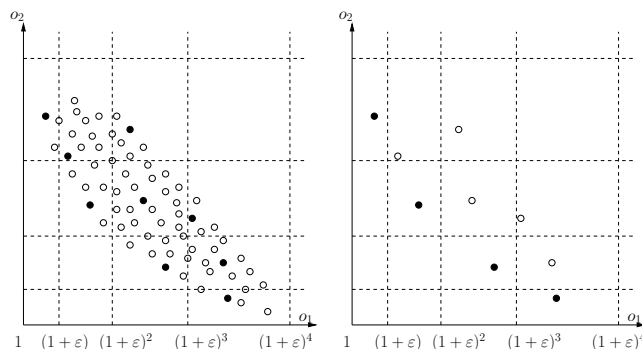


FIGURE 8 – Grille logarithmique.

Disposant de ces outils, des algorithmes d'approximation avec garantie de performance, et plus précisément des FPTAS, ont été proposés pour divers problèmes d'optimisation combinatoire multi-objectifs : plus court chemin multi-objectifs, arbre couvrant multi-objectifs, sac-à-dos multi-objectifs, certains problèmes d'ordonnancement multi-objectifs (Warburton, 1987; Papadimitriou et Yannakakis, 2000; Erlebach et al., 2002; Laumanns et al., 2002; Angel et al., 2003). Concernant le problème de plus court chemin multi-objectifs plus spécifiquement, plusieurs FPTAS ont été proposés dans la littérature, c'est-à-dire des algorithmes qui, étant donnés un encodage de l'instance et un niveau de précision $\varepsilon > 0$, retournent une ε -couverture en temps et en espace bornés par un polynôme de la taille de l'encodage et de $\frac{1}{\varepsilon}$. Hansen (1980) et Warburton (1987) ont proposé des méthodes utilisant des techniques de *rounding and scaling* (ce qui consiste à rendre plus grossière l'échelle de valuation afin de diminuer la complexité de l'algorithme) en partant d'algorithmes exacts pseudo-polynomiaux (c'est-à-dire qui opèrent en temps et en espace bornés par un polynôme de la taille de l'encodage et de la plus grande donnée de l'instance), dans le but d'avoir des données auxiliaires de taille polynomialement bornée lors de l'exécution. Ces méthodes sont propres aux problèmes bi-objectifs et aux graphes sans circuit respectivement. Un autre algorithme d'approximation est dû à Papadimitriou et Yannakakis (2000). Il est moins spécifique au plus court chemin multi-objectifs, et son principal intérêt réside dans sa généralité : on procède en calculant une solution (si elle existe) pour chaque case de la grille logarithmique. Les auteurs montrent que cela peut être réalisé en temps polynomial pour un problème multi-objectifs P si il existe un algorithme pseudo-polynomial pour la version mono-objectif exacte de P (étant donnée une instance de P et un entier K , existe-t-il une solution réalisable qui coûte exactement K ?). Enfin, Tsaggouris et Zaroliagis (2009) ont récemment proposé un FPTAS en partant d'une version multi-objectifs de l'algorithme de Bellman-Ford.

Hormis l'approche de Papadimitriou et Yannakakis (2000), toutes les autres propositions reposent sur la programmation dynamique. L'originalité de notre proposition concernant le développement d'un algorithme d'approximation pour le plus court chemin multi-objectifs est qu'elle s'appuie sur la version multi-objectifs de l'algorithme de Dijkstra, qui est un algorithme glouton dans son principe. Nous avons proposé deux variantes de cet algorithme pour déterminer une frontière ε -approchée, que nous ne détaillerons pas ici. L'une des variantes est un FPTAS et requiert de pouvoir borner la longueur des chemins (en nombre d'arcs), l'autre n'est pas un FPTAS mais donne des résultats numériques plus performants en pratique. De par le nombre limité de solutions requises pour obtenir une ε -couverture de la frontière de Pareto, le gain de temps comparativement à une approche exacte est significatif. En construisant des instances à dessein, l'écart entre la cardinalité de l'ensemble de Pareto et la cardinalité d'une ε -couverture peut devenir très important, comme illustré sur la figure 9. L'instance utilisée

a été proposée par Hansen (1980), et est construite en sorte que l'ensemble des points représentant les solutions réalisables sont distincts et alignés dans l'espace des objectifs (voir figure 10), et par conséquent toutes Pareto-optimales. A contrario, il y a seulement deux solutions Lorenz-optimales pour cette classe d'instances (celles situées de part et d'autre de la bissectrice du quadrant). Lorsque le choix de la dominance de Lorenz est pertinent (autrement dit qu'elle se justifie dans le contexte décisionnel étudié, comme par exemple lorsque les objectifs représentent les opinions de différents agents), cette dernière notion est donc utile pour limiter le nombre de solutions retournées. Nous développons cette idée dans la section suivante.

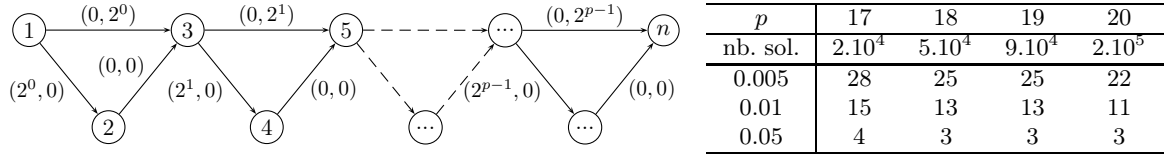


FIGURE 9 – Taille d'une ε -couverture pour l'instance de Hansen. L'instance de Hansen est représentée à gauche, avec $n = 2p + 1$. Le nombre de solutions Pareto-optimales est indiqué dans la deuxième ligne du tableau de droite, pour différentes valeurs de p , et la taille des ε -couvertures obtenues avec la grille logarithmique est indiquée pour $\varepsilon = 0.005, 0.01$ et 0.05 .

4.1.3 Détermination de la frontière de Lorenz

Nous présentons les résultats de cette section en minimisation.

Dans des problèmes combinatoires où les objectifs représentent les points de vue de différents agents, il paraît légitime de s'intéresser à la détermination des solutions Lorenz-optimales, tant d'un point de vue décisionnel (ce sont les solutions « équitables » en un certain sens) qu'algorithmique (ces solutions sont en nombre moins important que les solutions Pareto-optimales). Tout comme pour la détermination du front de Pareto, nous avons montré que le problème de décision associé à la détermination du front de Lorenz est NP-complet pour de nombreux problèmes classiques dont la version mono-objectif est polynomiale (problème de décision associé : étant donné un vecteur $y = (y_1, \dots, y_k)$, existe-t-il une solution qui domine y au sens de Lorenz?), comme par exemple pour l'arbre couvrant multi-objectifs et le plus court chemin multi-objectifs (Perny et Spanjaard, 2003; Perny et al., 2006). De plus, pas plus que pour le front de Pareto, on ne peut borner la cardinalité de la frontière de Lorenz par un polynôme de la taille de l'instance. En effet, en modifiant légèrement les classes d'instances pour lesquelles le nombre de points dans la frontière de Pareto s'accroît exponentiellement avec la taille des données (instances de Hansen), on peut construire des classes d'instances pour lesquelles la frontière de Lorenz grandit aussi exponentiellement (voir figure 11) : en effet, dans les nouvelles instances ainsi obtenues, toutes les solutions sont Lorenz-optimales, comme on pourra s'en convaincre facilement en examinant le cône de dominance de Lorenz représenté sur le graphe de droite de la figure.

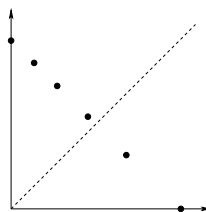


FIGURE 10 – Image des solutions réalisables dans l'instance de Hansen.

Toutefois, en pratique, le nombre d'éléments non-dominés au sens de Lorenz est souvent bien plus réduit que le nombre d'éléments non-dominés au sens de Pareto. Par exemple, dans des problèmes d'arbre couvrant multi-objectifs sur des cliques à 25 sommets (25^{23} arbres couvrants en tout), le nombre de points Pareto-optimaux est de l'ordre de 250 en moyenne pour 2 objectifs avec des coûts entiers tirés aléatoirement dans $[1, 100]$ (Sourd et al., 2006), quand que le nombre de solutions Lorenz-optimales est de l'ordre de 10 en moyenne pour 5 objectifs avec des coûts entiers tirés aléatoirement dans $[1, 1000]$ (Perny et al., 2006) (rappelons qu'au plus le nombre d'objectifs et la largeur de l'échelle sont importants, au plus il y a de points non-dominés potentiels). Cet écart de cardinalité se conçoit bien quand on réalise que le cône de dominance de Lorenz est beaucoup plus étendu que celui de Pareto (voir partie droite de la figure 11).

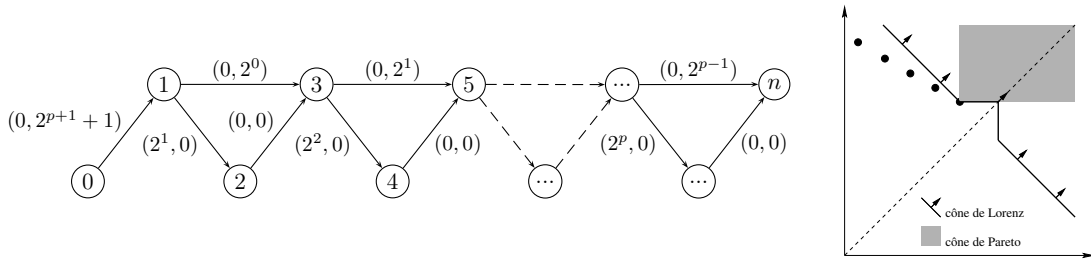


FIGURE 11 – Une instance où tous les chemins sont Lorenz-optimaux ($n = 2p + 1$).

La détermination de l'ensemble des solutions Lorenz-optimales peut se faire en exploitant la propriété suivante (Perny et Spanjaard, 2003) : étant donnés deux vecteurs y et y' , si $\sum_{i=1}^k y'_i > k \cdot y_{(k)}$ alors y domine y' au sens de Lorenz (rappelons que $y_{(k)}$ désigne la plus grande composante de y). Pour ce faire, on énumère les solutions $x^1, \dots, x^j$ dans l'ordre croissant de la somme de leurs composantes (sous réserve qu'un algorithme d'énumération des meilleurs solutions soit disponible pour la version mono-objectif du problème étudié), et on stoppe l'énumération lorsque l'on vérifie $\sum_{i=1}^k f_i(x^j) > k \cdot f_{(k)}(x^*)$, où x^* est la solution dont la plus grande composante est minimale dans $x^1, \dots, x^j$, et $f_{(k)}(x^*)$ est sa plus grande composante. Toutes les solutions énumérées ensuite seraient en effet dominées par x^* au sens de Lorenz, d'après la propriété précédente. Cette forme de procédure d'énumération implicite peut également être utilisée pour la recherche d'une solution de meilleur compromis au sens d'une fonction d'agrégation, comme nous allons le voir dans la section suivante.

4.2 Détermination d'une solution de meilleur compromis

Depuis les travaux précurseurs de Steuer et Choo (1983) pour développer une méthode interactive en programmation mathématique multi-objectifs, l'approche par détermination d'une solution de compromis a été étudiée pour divers problèmes d'optimisation combinatoire et diverses fonctions d'agrégation, telles que le critère min max (Murthy et Her, 1992; Punnen et Aneja, 1995; Yu, 1998), le critère leximax (Ehrgott et Skriver, 2003), ou encore la distance de Tchebycheff au point idéal (Granat et Guerriero, 2003; Galand et Perny, 2006).

Un problème générique de recherche d'une solution de compromis peut se formuler à l'aide d'un programme mathématique. En notant $\varphi : \mathbb{R}^k \rightarrow \mathbb{R}$ la fonction d'agrégation multi-objectifs, un problème de recherche d'une solution de compromis peut s'écrire comme suit :

$$(P) \quad \begin{cases} \text{opt } \varphi(y) \\ \text{s.t. } y = f(x) \\ x \in \mathcal{X} \end{cases}$$

Dans l'espace des objectifs, le problème P peut se reformuler simplement comme $\text{opt}_{y \in \mathcal{Y}} \varphi(y)$, où $\mathcal{Y} = f(\mathcal{X})$. La difficulté de résolution de P vient de la nature souvent non-linéaire de la fonction φ , qui conduit à des problèmes NP-difficiles. Comme d'usage en recherche opérationnelle face à des problèmes NP-difficiles, la plupart des méthodes de résolution exacte de P repose donc sur l'obtention de bornes –calculables efficacement– sur la valeur de la solution optimale. Afin d'obtenir ces bornes, nous distinguons deux types de relaxations :

- *Relaxation de l'ensemble image* : cela consiste à définir un ensemble $\mathcal{Y}' \supseteq \mathcal{Y}$ et à résoudre le problème $P_{\mathcal{Y}'}$ défini par :

$$(P_{\mathcal{Y}'}) \quad \begin{cases} \text{opt} & \varphi(y) \\ \text{s.c.} & y \in \mathcal{Y}' \end{cases}$$

- *Relaxation de la fonction objectif* : cela consiste à définir une fonction $\varphi' : \mathbb{R}^k \rightarrow \mathbb{R}$ telle que pour tout $y \in \mathcal{Y}$, $\varphi'(y) \leq \varphi(y)$ (ou $\varphi'(y) \geq \varphi(y)$, selon le sens d'optimisation) et à résoudre le problème $P_{\varphi'}$ défini par :

$$(P_{\varphi'}) \quad \begin{cases} \text{opt} & \varphi'(y) \\ \text{s.c.} & y = f(x) \\ & x \in \mathcal{X} \end{cases}$$

La problématique principale est alors de définir $\mathcal{Y}'$ (resp. φ') de sorte que la valeur optimale du problème $P_{\mathcal{Y}'}$ (resp. $P_{\varphi'}$) puisse être calculée efficacement et fournir une borne la plus fine possible. Dans cette optique, il semble opportun de considérer des relaxations continues de $\mathcal{Y}$ (resp. des fonctions φ' linéaires). Nous avons proposé des procédures de calcul de bornes fondées sur ces deux types de relaxations. Leur efficacité computationnelle dépend de la capacité à résoudre efficacement la version mono-objectif du problème P étudié. Par conséquent, les bornes obtenues peuvent être exploitées pour résoudre un large éventail de problèmes multi-objectifs dont la version mono-objectif peut être résolue efficacement. Nous donnons ci-dessous deux exemples : un exemple de relaxation de l'ensemble image pour borner la valeur optimale pour la moyenne ordonnée pondérée (Galand et Spanjaard, 2007a,b, 2012), un exemple de relaxation de la fonction objectif pour borner la valeur optimale pour l'intégrale de Choquet (Galand et al., 2009, 2010).

4.2.1 Relaxation de l'ensemble image

Nous présentons les résultats de cette section en minimisation.

Nous présentons ici un exemple de procédure pour calculer une borne de la valeur optimale de P pour $\varphi = \text{OWA}$, obtenue par relaxation de l'ensemble image. Pour cela, on définit une relaxation $\mathcal{Y}'$ et un programme $P_{\mathcal{Y}'}$ comme suit :

$$\mathcal{Y}' = \{y \in \mathbb{R}^k : y_i \geq b_i \forall i = 1, \dots, k \text{ et } \sum_{i=1}^k y_i \geq b_0\}$$

où $b_i = \min\{f_i(x) : x \in \mathcal{X}\}$ est la valeur d'une solution optimale selon f_i , et $b_0 = \sum_{i=1}^k f_i$. Les valeurs b_i ($i = 1, \dots, k$) sont obtenues en lançant n fois un algorithme standard pour la version mono-objectif du problème étudié (dès lors qu'il peut être efficacement résolu) pour les différentes valuations f_i successivement ($i = 1, \dots, k$).

Comme indiqué plus haut, pour que la relaxation soit intéressante en pratique, il faut disposer d'une procédure efficace pour calculer la valeur optimale de $P_{\mathcal{Y}'}$. Lorsque la fonction objectif est une moyenne ordonnée pondérée $\sum_{i=1}^k \lambda_i y_{(i)}$, où $y_{(1)} \leq y_{(2)} \leq \dots \leq y_{(k)}$, la résolution de $P_{\mathcal{Y}'}$ nécessite bien sûr de prendre en compte la façon dont les composantes

de y s'ordonnent. La moyenne ordonnée pondérée est une fonction d'agrégation linéaire par morceaux. Plus précisément, elle est linéaire dans chaque sous-espace *comonotone* de vecteurs, c'est-à-dire un sous-espace tel que pour chaque couple y, y' de vecteurs, il existe une même permutation π de $(1, \dots, k)$ pour laquelle $y_{\pi(1)} \leq \dots \leq y_{\pi(k)}$ et $y'_{\pi(1)} \leq \dots \leq y'_{\pi(k)}$. Le problème $P_{\mathcal{Y}'}$ peut donc être divisé en plusieurs sous-problèmes, un pour chaque sous-espace comonotone de $\mathbb{R}^k$. La résolution de $P_{\mathcal{Y}'}$ revient alors à résoudre chacun des programmes linéaires $P_{\mathcal{Y}', \pi}$ définis pour chaque permutation π particulière de $(1, \dots, k)$:

$$(P_{\mathcal{Y}', \pi}) \quad \begin{cases} \min \sum_{i=1}^k \lambda_i y_{\pi(i)} \\ \text{s.c.} & y_{\pi(i)} \leq y_{\pi(i+1)} \quad \forall i \in \{1, \dots, k-1\}, \\ & y_i \geq b_i \quad \forall i \in \{1, \dots, k\}, \\ & \sum_{i=1}^k y_i \geq b_0, \\ & y \in \mathbb{R}^k. \end{cases}$$

La valeur d'une solution optimale $y_{\mathcal{Y}'}$ de $P_{\mathcal{Y}'}$ est $\min_{\pi \in \Pi} \text{OWA}(y_{\mathcal{Y}', \pi}^*)$ où $y_{\mathcal{Y}', \pi}^*$ est une solution optimale du programme linéaire $P_{\mathcal{Y}', \pi}$ et Π l'ensemble des toutes les permutations possibles. Pour k objectifs, il y a donc $|\Pi|=k!$ programmes linéaires à résoudre. Néanmoins, en pratique, il n'est pas nécessaire de résoudre les $k!$ programmes linéaires. Il existe en effet une permutation facilement calculable π^* pour laquelle $\text{OWA}(y_{\mathcal{Y}'}^*) = \text{OWA}(y_{\mathcal{Y}', \pi^*}^*)$: c'est la permutation π^* pour laquelle $b_{\pi^*(1)} \leq b_{\pi^*(2)} \leq \dots \leq b_{\pi^*(k)}$ (Galand et Spanjaard, 2007b). Si les λ_i sont croissants (ce qui, en minimisation, permet de favoriser les profils équilibrés), le programme linéaire $P_{\mathcal{Y}', \pi^*}$ peut être résolu en temps linéaire par une procédure gloutonne que nous ne détaillerons pas ici (le lecteur intéressé pourra consulter Galand et Spanjaard, 2007b), et ainsi fournir une borne. Nous donnons ci-dessous un exemple numérique. Précisons toutefois que, même si la nature des λ_i permet parfois d'accélérer la résolution de $P_{\mathcal{Y}', \pi^*}$, la valeur $\text{OWA}(y_{\mathcal{Y}', \pi^*}^*)$ reste bien sûr une borne pour tout jeu de poids λ .

Exemple 9 *Supposons que l'on cherche à borner la valeur optimale pour le nuage de points représenté sur la figure 12. Soient $b = (5, 10)$ et $b_0 = 17$. L'ensemble $\mathcal{Y}'$ est la zone de l'espace délimitée par les contraintes en gras. Comme $b_1 \leq b_2$, on a $\pi^*(1) = 1$ et $\pi^*(2) = 2$: la solution optimale se trouve donc dans la zone grisée sur la figure. Résoudre $P_{\mathcal{Y}'}$ revient ainsi à résoudre le programme $P_{\mathcal{Y}', \pi^*}$:*

$$\begin{cases} \min & \lambda_1 y_1 + \lambda_2 y_2 \\ \text{s.c.} & y_1 \leq y_2, \\ & y_1 \geq 5 \quad y_2 \geq 10, \\ & y_1 + y_2 \geq 17, \\ & y_1 \in \mathbb{R}, y_2 \in \mathbb{R}. \end{cases}$$

Si $\lambda_1 < \lambda_2$, la procédure gloutonne de résolution retourne $y_{\pi^*(1)}^* = y_1^* = 7$ et $y_{\pi^*(2)}^* = y_2^* = 10$. La borne qui s'ensuit est bien sûr $7\lambda_1 + 10\lambda_2$.

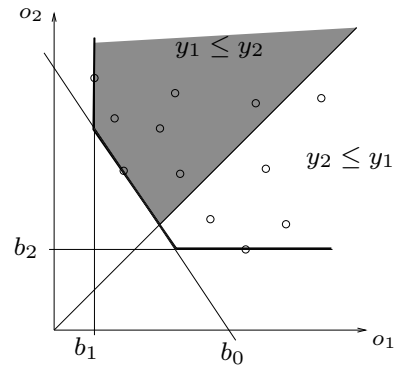


FIGURE 12 – Sous-espaces comonotones.

4.2.2 Relaxation de la fonction objectif

Nous présentons les résultats de cette section en maximisation.

La seconde façon de calculer une borne pour P est de procéder par relaxation de la fonction

objectif. Nous illustrons cette seconde façon de procéder en montrant comment obtenir une borne calculable efficacement pour l'intégrale de Choquet, en se plaçant dans un contexte de maximisation. Les résultats sont toutefois transposables en minimisation. Pour borner supérieurement l'intégrale de Choquet telle que présentée en section 3.1.1, il nous faut faire appel à la notion de *cœur* d'une capacité. Notons tout d'abord qu'à toute capacité v on peut associer une capacité *duale* $\bar{v}$ définie par $\bar{v}(A) = 1 - v(N \setminus A)$ pour tout $A \subseteq N$. Remarquons que la capacité biduale $\bar{\bar{v}}$ est égale à v . De plus, quand v est supermodulaire, on a $v(A) + v(N \setminus A) \leq 1$, et donc $v(A) \leq \bar{v}(A)$. Il fait alors sens de définir le *cœur* d'une capacité v comme suit : $\text{cœur}(v) = \{\lambda \in \mathcal{L} : v(A) \leq \lambda(A) \leq \bar{v}(A)\}$, où $\mathcal{L}$ est l'ensemble des capacités *additives* (c'est-à-dire que la capacité d'un ensemble est égale à la somme des poids de ses singletons) définies sur 2^N .

Lorsque v est supermodulaire, Shapley (1971) a montré que le cœur de v est non-vidé. Schmeidler (1986) (voir aussi Schmeidler, 1989) a fourni une interprétation intuitive de l'intégrale de Choquet dans ce cas : on peut la voir comme une somme pondérée avec des poids imprécis, où les jeux de poids possibles sont l'ensemble des jeux de poids du cœur. Plus précisément, pour un vecteur y donné, la valeur de l'intégrale de Choquet $C_v(y)$ est le minimum de toutes les valeurs possibles des sommes pondérées dont les poids appartiennent au cœur de la capacité. Ce résultat est aussi lié aux travaux d'Edmonds en optimisation sous-modulaire (Edmonds, 1970; Fujishige, 2005), aspect que nous ne détaillerons pas ici. Une conséquence très utile de ce résultat est que, pour toute capacité additive $\lambda \in \text{cœur}(v)$, on a $C_v(y) \leq \sum_{i=1}^k \lambda_i y_i$ pour $\lambda \in \text{cœur}(v)$. L'idée est alors d'utiliser cette fonction linéaire $\varphi'_\lambda(y) = \sum_{i=1}^k \lambda_i y_i$ comme relaxation de la fonction objectif. La linéarité de φ' permet de calculer la borne en déterminant une solution optimale du problème mono-objectif obtenu en scalarisant chaque valuation par φ'_λ .

Exemple 10 La table 4 donne une capacité v sur trois objectifs, sa duale $\bar{v}$, et une capacité additive λ dans $\text{cœur}(v)$. Pour $y = (7, 10, 3)$ par exemple, on vérifie bien que $C_v(y) = 6.1 \leq 6.4 = \varphi'_\lambda(y)$.

	$\emptyset$	$\{1\}$	$\{2\}$	$\{3\}$	$\{2, 3\}$	$\{1, 3\}$	$\{1, 2\}$	N
v	0	0.4	0.1	0.2	0.3	0.8	0.7	1
λ	0	0.5	0.2	0.3	0.5	0.8	0.7	1
$\bar{v}$	1	0.7	0.2	0.3	0.6	0.9	0.8	1

TABLE 4 – Capacité additive dans le cœur.

Une question qui survient naturellement est le choix d'un vecteur de poids du cœur pour la borne, puisque chaque vecteur de poids du cœur fournit une borne. Afin d'obtenir la meilleure borne possible, on peut résoudre le programme suivant :

$$\begin{aligned}
 & \min_{\lambda \in \mathbb{R}^k} g(\lambda) \\
 \text{s.t. } & \sum_{i \in A} \lambda_i \geq v(A) \quad \forall A \subseteq N, \\
 & \sum_{i \in Q} \lambda_i = 1, \\
 & \lambda_i \geq 0 \quad \forall i = 1, \dots, k.
 \end{aligned}$$

où $g(\lambda)$ désigne la valeur de la solution optimale pour des valuations scalarisées selon φ'_λ . Comme g est une fonction linéaire par morceaux (car c'est l'enveloppe inférieure d'un ensemble

de fonctions linéaires, une par jeu de poids dans le cœur), on peut résoudre le programme en faisant appel à un algorithme de type sous-gradient. Plus précisément, dans nos travaux, nous avons fait appel au r -algorithme de Shor, qui est disponible dans la librairie SolvOpt (Kappel et Kuntsevich, 2000). Remarquons que le résultat de ce programme donne une borne même si la capacité v n'est pas supermodulaire. Toutefois, l'optimum peut alors correspondre à un jeu de poids λ ne sommant pas à un 1, et la borne obtenue est donc assez grossière. Nous avons exploité les types de bornes présentés dans cette section et la précédente pour développer des algorithmes de résolution pour divers problèmes combinatoires. Nous décrivons ces mises en œuvre dans la suite, mais au préalable nous présentons les différents types d'algorithmes d'énumération implicite dans le cadre desquels ces bornes peuvent être exploitées.

4.2.3 Algorithmes d'énumération implicite pour la recherche de compromis

Nous présentons les résultats de cette section en minimisation.

Les algorithmes d'énumération implicite pour la détermination d'une solution de meilleur compromis peuvent être classés en quatre catégories, selon la manière dont l'espace des solutions réalisables est exploré.

1. *Approche par rangement* : le principe de cette approche, initialement proposée par Hamacher et Ruhe (1994) pour déterminer un arbre couvrant minimax dans un problème multi-objectifs, est d'énumérer les meilleures solutions $x^1, \dots, x^j$ dans l'ordre croissant d'une somme pondérée des objectifs telle que $\sum_i \lambda_i y_i \leq \varphi(y)$, et de stopper l'énumération lorsque l'on vérifie $\sum_i \lambda_i f_i(x^j) \geq \varphi(f(x^*))$ pour la solution x^j courante, où x^* désigne la meilleure solution trouvée jusqu'alors. La réalisation de cette condition garantit en effet que x^* est la solution minimisant $\varphi(f(x))$ (dans le cas d'une maximisation on énumère bien sûr dans l'ordre décroissant, et la condition est inversée). Cette approche peut s'appliquer à tout problème multi-objectifs pour lequel un algorithme de rangement efficace existe pour la version mono-objectif, comme par exemple : arbre couvrant (Gabow, 1977), plus court chemin (Eppstein, 1998), couplage (Chegireddy et Hamacher, 1987), flot (Hamacher, 1995) et divers problèmes d'ordonnancement (Brucker et Hamacher, 1989).
2. *Procédures de branch and bound* : ces procédures bien connues présentent l'intérêt d'être moins gourmandes en mémoire vive que l'approche précédente. En effet, les algorithmes de rangement procèdent généralement par une exploration en largeur de l'espace des solutions, ce qui peut conduire à des dépassements des ressources en mémoire vive lorsque la taille des instances augmente. Les branch and bounds avec stratégie d'exploration en profondeur d'abord évitent cet écueil, et permettent donc de résoudre des problèmes de tailles plus importantes, même si ils sont parfois moins rapides que l'approche par rangement sur les instances de taille modérée. Cette méthode présente en outre l'avantage d'être très générale et de permettre d'exploiter les stratégies de branchement développées pour le cadre mono-objectif. L'un des premiers travaux dans cette direction est dû à Chung et al. (1993), qui proposent une procédure de branch and bound relativement générique pour résoudre des problèmes multi-objectifs où l'on cherche à déterminer une solution x minimax, c'est-à-dire minimisant $\varphi(f(x)) = \max\{f_1(x), f_2(x)\}$. En chaque nœud de ce branch and bound, la borne supérieure calculée est $\min\{\varphi(f(x_1^*)), \varphi(f(x_2^*))\}$, où x_i^* est la solution optimale pour f_i dans le sous-problème considérée. La borne inférieure proposée est $\max\{g(\lambda) : \sum_i \lambda_i = 1, \lambda_i \geq 0 \forall i\}$, où $g(\lambda)$ désigne la valeur de la solution x minimisant $\sum_i \lambda_i f_i(x)$, ce qui peut être vu comme une spécification au cas minimax de la borne donnée dans la section précédente pour l'intégrale de Choquet.
3. *Algorithmes par développements d'étiquettes* : les approches par développements d'éti-

quettes visent à raffiner la programmation dynamique multi-objectifs. Cette dernière permet en effet d'énumérer les solutions Pareto-optimales, parmi lesquelles se trouvent nécessairement une solution optimale pour φ (dès lors que cette fonction est compatible avec la dominance de Pareto). La programmation dynamique multi-objectifs s'appuie sur les équations de Bellman, qui s'écrivent sous la forme suivante dans leur version multi-objectifs (une équation par sommet i du graphe d'états) :

$$\Pi_i = \min_{j \in \text{Pred}(i)} \{f(j, i) + y : y \in \Pi_j\}$$

où Π_i désigne l'image –dans l'espace des objectifs– des solutions Pareto-optimales au sommet i , $\text{Pred}(i)$ l'ensemble de ses prédécesseurs, $f(j, i)$ le vecteur-coût assigné à l'arc (j, i) et $\min$ l'opération de sélection des points non-dominés d'un ensemble. L'idée est d'exploiter des bornes pour supprimer par avance des étiquettes qui ne pourront pas conduire à une solution optimale, et ainsi accélérer la recherche en réduisant le cardinal des ensembles Π_i . Un des premiers travaux employant cette approche en optimisation multi-objectifs est dû à Murthy et Her (1992), afin de déterminer un chemin minimax dans un graphe. À titre d'illustration, considérons la figure 13 où l'on cherche le chemin x minimisant $\varphi(f(x)) = \max\{f_1(x), f_2(x)\}$ de 1 à 3. Pour obtenir une borne, on utilise la fonction linéaire $\varphi'(f(x)) = 0.5f_1(x) + 0.5f_2(x)$ (on a en effet clairement $\varphi'(y) \leq \varphi(y)$). Supposons qu'on ait prétraité l'instance en calculant les plus courts chemins selon φ' de 3 à tous les autres sommets dans le graphe inversé, et que le chemin de valeur $(7, 4)$ de 1 à 3 est connu. L'étiquette $(4, 6)$, non-dominée au sommet 2, peut alors être supprimée car $\varphi'(4, 6) + 2.5 = 7.5 > 7 = \varphi(7, 4)$, autrement dit cette étiquette ne pourra conduire à une solution optimale pour φ . Pour finir, notons que les méthodes de recherche heuristique multi-objectifs d'une solution de meilleur compromis (Perny, 2000; Mandow et de la Cruz, 2003; Galand et Perny, 2006) entrent dans cette catégorie d'algorithmes.

4. *Formulations en programmation linéaire en variables mixtes* (dite aussi formulation MIP, pour *Mixed Integer Programming*) : cela consiste tout d'abord à formuler le problème d'optimisation combinatoire étudié sous la forme d'un programme linéaire en variables mixtes, la contribution originale à cette étape se situant alors généralement dans la linéarisation de la fonction objectif (de type OWA ou Choquet par exemple). Les contributions algorithmiques peuvent être ensuite de différentes natures : prétraitement de l'instance afin de réduire le temps de calcul pour la résolution proprement dite, renforcement de la formulation afin de s'approcher de l'enveloppe convexe des points entiers...

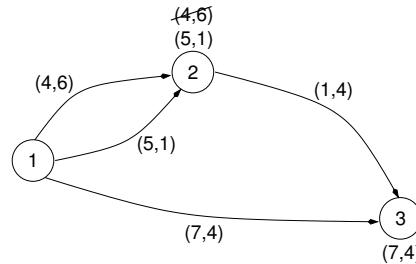


FIGURE 13 – L'étiquette $(4,6)$ est supprimée à l'aide de la borne.

Nous avons adopté des approches de ces différents types dans nos travaux. Nous en dressons un panorama dans la section suivante.

4.2.4 Mises en œuvre

Les bornes précédemment décrites ont été exploitées en particulier pour proposer des méthodes de résolution pour la recherche de compromis dans les problèmes de l'arbre couvrant multi-objectifs, du plus court chemin multi-objectifs et du sac à dos multi-objectifs. Le tableau 5 présente une synthèse de ces contributions, indiquant pour chaque problème le type des méthodes proposées. L'approche de compromis nous a permis d'aborder des problèmes comportant de 3 à 10 objectifs, ce qui est difficilement envisageable avec l'approche Pareto. À notre sens, l'apport principal de ces travaux est d'offrir des solutions algorithmiques pour "dépasser" le modèle minimax (la plupart des travaux visant à déterminer une unique solution de compromis dans des problèmes multi-objectifs se focalisaient sur la recherche d'une solution minimax, ou maximin) et optimiser des fonctions d'agrégation plus sophistiquées, telles que la moyenne ordonnée pondérée (OWA) ou l'intégrale de Choquet.

	AC	PCC	SD
OWA	- rangement (Galand et Spanjaard, 2007a) - branch and bound (Galand et Spanjaard, 2007a) - formulation MIP (Galand et Spanjaard, 2012)	développement d'étiquettes (Galand et Spanjaard, 2007b)	
Choquet	- rangement (Galand et al., 2010) - branch and bound (Galand et al., 2009) (Galand et al., 2010)	- rangement (Galand et al., 2010) - développement d'étiquettes (Galand et al., 2010) - formulation MIP (Galand et al., 2011)*	- branch and bound (Galand et al., 2009) - formulation MIP (Lesca et Perny, 2010)* (Galand et al., 2011)*

TABLE 5 – Synthèse des contributions sur la recherche de compromis selon OWA et Choquet. Je suis coauteur de toutes les références indiquées, y compris pour Choquet (si si, dans le "et al." !). Je ne suis toutefois pas coauteur des deux références marquées d'une astérisque, qui complètent le panorama des méthodes de résolution. AC : Arbre Couvrant, PCC : Plus Court Chemin, SD : Sac à Dos.

Arbre couvrant multi-objectifs. Sur ce problème, nous avons étudié l'optimisation de la moyenne ordonnée pondérée (Galand et Spanjaard, 2007b, 2012) et de l'intégrale de Choquet (Galand et al., 2009, 2010). Nous décrivons ici brièvement les travaux conduits sur l'optimisation de la moyenne ordonnée pondérée. Nous avons développé d'une part des branch and bounds reposant sur les deux types de bornes présentées précédemment (Galand et Spanjaard, 2007b), d'autre part une approche MIP (Galand et Spanjaard, 2012) s'appuyant sur une formulation classique de l'arbre couvrant minimum via des contraintes de flot (voir Magnanti et Wolsey, 1995). Avec cette dernière approche, nous traitons non seulement le cas des poids croissant avec les valeurs des composantes (favorisant les profils équilibrés), mais aussi le cas des poids quelconques. En effet, nous exploitons deux linéarisations de la fonction OWA qui ont été proposées dans la littérature : une linéarisation proposée par Ogryczak et Sliwinski (2003) qui est valide uniquement pour des poids croissants, et une linéarisation proposée par Boland et al. (2006) dans le cadre du *discrete ordered median problem* (où l'on cherche un sous-ensemble de k sommets d'un graphe où positionner des services de façon à satisfaire les clients situés en chaque sommet, la satisfaction d'un client étant égale à sa distance au service le plus proche) qui est elle valide pour tout jeu de poids, mais qui nécessite en contrepartie l'introduction de variables binaires supplémentaires. Notre contribution se situe principalement

dans le développement de procédures de prétraitement qui permettent de réduire de façon importante la taille de l'instance, la validité de ces procédures dépendant encore une fois de la nature du jeu de poids utilisé dans la fonction OWA (croissant ou quelconque). Ces procédures sont de deux types : une généralisation des règles de cycle et de cocycle au cas de l'optimisation d'un OWA avec poids croissants, et des procédures de *shaving* valides pour tout jeu de poids. Le *shaving* consiste ici à calculer une borne sur la valeur optimale lorsqu'on rend une arête obligatoire (resp. interdite), et comparer cette borne à la valeur OWA d'une solution connue (calculée à l'aide d'une heuristique) pour éventuellement fixer cette arête comme interdite (resp. obligatoire). Des tests numériques ont été conduits pour différents jeux de poids. Il y a bien évidemment deux facteurs de difficulté : le nombre de sommets du graphe et le nombre d'objectifs (fixé à trois au moins, puisque l'on dispose d'un algorithme efficace pour déterminer la frontière de Pareto dans son entier dans le cas bi-objectifs). Pour des poids croissants, l'approche couplant prétraitements et solveur permet de résoudre des instances comportant jusqu'à 100 sommets ou jusqu'à 8 objectifs (sur un Core3 3.0Ghz, avec 3Go de RAM). Plus précisément, pour 3 objectifs et un jeu de poids éloigné de la moyenne arithmétique, on arrive à déterminer un arbre couvrant optimal pour OWA en moins de 15 minutes en moyenne (toutes phases comprises) sur des graphes complets comportant jusqu'à 100 sommets avec des coûts tirés aléatoirement dans $[1, 100]$ selon une loi uniforme ; pour 5 objectifs on s'arrête à 60 sommets, et pour 8 objectifs à 35 sommets. Pour des poids quelconques, les résultats numériques obtenus sont bien sûr plus modestes : dans la limite de temps de 15 minutes, on peut traiter des instances comportant jusqu'à 50 sommets et/ou 5 objectifs (à moduler selon la nature exacte du jeu de poids utilisé). Ces résultats constituent néanmoins un premier pas vers la résolution de ce problème lorsque les poids sont quelconques.

Plus court chemin multi-objectifs. Nos travaux sur la recherche de compromis dans le problème de plus court chemin multi-objectifs exploitent les mêmes types de bornes que ceux sur l'arbre couvrant multi-objectifs, tant pour OWA (Galand et Spanjaard, 2007b) que pour Choquet (Galand et al., 2010), mais diffèrent dans la nature des algorithmes employés. En effet, si l'approche par rangement et les formulations MIP restent des alternatives algorithmiques de choix dans les problèmes de chemin, l'approche par développement d'étiquettes s'impose plus naturellement que le branch and bound, de par la possibilité offerte d'exploiter les résultats obtenus en programmation dynamique multi-objectifs (ce qui n'était pas le cas pour le problème de l'arbre couvrant multi-objectifs). Pour un nombre réduit d'objectifs, l'approche par développement d'étiquettes est plus performante que l'approche par rangement, vraisemblablement du fait que les ensembles d'étiquettes restent de taille raisonnable. Pour les deux approches, les temps de résolution (sur un Core2 2.66Ghz, avec 2Go de RAM) restent au pire de l'ordre de quelques secondes (plus souvent inférieurs à la seconde) sur des graphes orientés comportant jusqu'à 4000 nœuds et jusqu'à 10 objectifs, graphes générés aléatoirement avec une densité d'arcs autour de 50% (la présence d'un arc est tirée aléatoirement avec une probabilité de 0.5). Le comportement diffère toutefois quelque peu sur les graphes en couches, avec certaines sous-classes de ces graphes pour lesquelles les temps de résolution peuvent augmenter de façon significative. Les performances numériques des algorithmes par développement d'étiquettes ont récemment commencées à être comparées avec celles obtenues en passant par une formulation MIP, pour la recherche de compromis guidée par l'intégrale de Choquet (Galand et al., 2011). La formulation MIP étudiée s'appuie sur une linéarisation de l'intégrale de Choquet fondée sur le résultat de Schmeidler $C_v(y) = \min_{\lambda \in \text{coeur}(v)} \sum_{i=1}^k \lambda_i y_i$, le cœur étant défini par un ensemble de contraintes linéaires. Les résultats numériques présentés tendent à montrer que l'usage d'une formulation MIP est une alternative prometteuse à l'approche par programmation dynamique, avec des résultats contrastés selon la famille d'ins-

tances envisagée (divers familles de graphes en couches sont étudiées, ainsi que divers familles de capacités supermodulaires dans l'intégrale de Choquet). L'approche par développement d'étiquettes est toutefois adaptable au cas des capacités quelconques, ce qui est moins le cas de la formulation MIP, où la linéarisation de l'intégrale de Choquet qui est utilisée n'est valide que pour les capacités supermodulaires.

Sac à dos multi-objectifs. Nous avons proposé et étudié un branch and bound pour la recherche d'un sac à dos optimal pour l'intégrale de Choquet (Galand et al., 2009). Avec cette approche, pour divers types de capacités dans l'intégrale de Choquet, on parvient à résoudre des instances comportant jusqu'à 150 objets et jusqu'à 8 objectifs, dans un délai variant de 1 seconde à 15 minutes (selon le nombre d'objets et le nombre d'objectifs). Après reformulation du problème en plus long chemin multi-objectifs dans un graphe en couches, une approche par développement d'étiquettes peut également être employée. Les résultats obtenus sont comparables avec ceux du branch and bound, mais les ressources en mémoire nécessaires sont beaucoup plus importantes, ce qui empêche parfois la résolution d'être menée à son terme (soit parce que les ressources sur la machine sont insuffisantes, soit parce que les "swaps" sur le disque dur ralentissent trop considérablement la procédure). Galand et al. (2011) ont comparé cette dernière approche avec une approche par MIP, la résolution par MIP s'avérant beaucoup plus performante. Toutefois, comme mentionné précédemment pour le problème de plus court chemin multi-objectifs, ce constat se limite au cas des capacités supermodulaires, puisque c'est dans ce cas que l'on dispose d'une linéarisation de l'intégrale de Choquet. Concernant cette problématique de linéarisation, signalons que la linéarisation via le résultat de Schmeidler conduit à un nombre de contraintes qui augmente exponentiellement avec le nombre d'objectifs (il y a autant de contraintes que de sous-ensembles d'objectifs). Un axe de recherche potentiellement fructueux est donc de viser à proposer des linéarisations plus "compactes", c'est-à-dire telles que le nombre de variables et de contraintes s'accroisse raisonnablement avec le nombre d'objectifs. Des travaux dans cette direction ont été initiés par Lesca et Perny (2010), qui proposent une formulation compacte dédiée au cas des capacités k -additives (Grabisch, 1997).

La fin de cette section achève la présentation de nos contributions en optimisation combinatoire multi-objectifs. Dans la prochaine section, nous donnons un aperçu de nos contributions en optimisation dans l'incertain. Ces deux facettes de nos travaux ne sont toutefois pas totalement déconnectées. D'une part, comme nous le soulignons dans la suite, l'optimisation combinatoire robuste, dans le modèle à scénarios discrets, a des liens très forts avec l'optimisation combinatoire multi-objectifs (elle peut être vue comme un cas particulier où tous les objectifs sont parfaitement commensurables). D'autre part, un parallèle certain existe entre la recherche de compromis en optimisation combinatoire multi-objectifs et la recherche d'une stratégie optimale en décision séquentielle dans l'incertain. En effet, bien que différents dans leurs structures, les problèmes algorithmiques soulevés sont parfois voisins. Dans les deux cadres, on dispose d'une fonction d'évaluation des solutions potentielles qui est souvent non-linéaire, du moins dans nos travaux, ce qui rend caduque une approche directe par programmation dynamique. A contrario, les préférences induites par ces fonctions d'évaluation sont compatibles avec des relations de dominance plus faibles (dominance de Pareto dans le cas multi-objectifs), qui sont elles conciliables avec la programmation dynamique, ce qui permet de s'y ramener malgré tout (comme dans les approches par développement d'étiquettes). Les réflexions conduites dans les deux cadres s'enrichissent donc mutuellement, dans un cycle vertueux de fertilisations croisées (si le lecteur nous permet cette métaphore biologique...).

5 Aspects computationnels de la décision dans l'incertain

La prise en compte explicite de l'incertitude dans la modélisation d'un problème permet d'une part de spécifier plus finement l'état de connaissance sur les conséquences des différents choix, d'autre part de s'offrir une palette de comportements de choix diversement risqués. L'un des premiers travaux sur le sujet est dû à Dantzig (1955), qui s'est intéressé à des problèmes de transport⁴ où la demande aux destinations est aléatoire, et qui a proposé une méthode de résolution fondée sur la programmation linéaire. C'est d'ailleurs d'abord plus particulièrement dans le domaine de la programmation mathématique que la prise en compte de l'incertain en optimisation a été étudiée, sous la dénomination de *programmation stochastique* (pour un état de l'art, consulter Birge, 1997; Sahinidis, 2004). Cet intitulé couvre différents paradigmes, parmi lesquels notamment :

- la programmation stochastique à deux étapes, où certaines variables sont fixées préalablement à l'occurrence d'événement(s) incertain(s), et d'autres postérieurement ; les variables fixées postérieurement peuvent l'être dans le but de rendre la solution réalisable ou de l'optimiser davantage ;
- la programmation stochastique robuste, où l'on insère un terme additionnel dans la fonction objectif du modèle précédent afin de tenir compte de l'attitude vis-à-vis du risque ; le terme additionnel est une mesure de variabilité, telle que la variance par exemple ;
- la programmation probabiliste, où la matrice des contraintes comporte des aléas, et où l'on fixe comme contrainte que la probabilité de faisabilité de la solution dépasse un certain seuil.

Toutes ces approches relèvent d'une prise en compte de l'incertitude où l'on change la structure des solutions réalisables. Par exemple, on peut considérer un problème de couplage maximum stochastique en deux étapes, où chaque arête e est munie de $k + 1$ fonctions poids $w_i : E \mapsto \mathbb{R}$, $i = 0, \dots, k$. Chaque arête $e \in E$ a un poids fixé $w_0(e)$ si elle est sélectionnée lors de la première étape. Par la suite, un des k scénarios possibles se réalise, disons i , avec une probabilité p_i . Le poids d'une arête lors de la deuxième étape si le scénario i s'est réalisé est $w_i(e)$. Une solution réalisable consiste à choisir un ensemble E_0 (éventuellement vide) d'arêtes à la première étape, puis de compléter cet ensemble E_0 en une solution réalisable en sélectionnant un ensemble E_i ($1 \leq i \leq k$) d'arêtes lors de la deuxième étape. Plus formellement, une solution réalisable est une collection d'ensembles d'arêtes $(E_0, E_1, \dots, E_k)$, et non plus un simple couplage. Cette approche de prise en compte de l'incertitude, bien qu'intéressante, est celle que nous avons le moins explorée dans nos travaux. Par intérêt personnel pour le soucis de procrastination qui s'exprime dans le paradigme de programmation stochastique en deux étapes (Immorlica et al., 2004), nous avons toutefois étudié ce problème de couplage maximum stochastique en deux étapes. Le lecteur intéressé pourra se reporter à Escoffier et al. (2010), article dans lequel nous présentons en particulier un algorithme d'approximation polynomiale de rapport $k/(2k - 1)$ pour la détermination d'une solution d'espérance maximale, ainsi que des algorithmes de complexité polynomiale (procédant par programmation dynamique) sur certaines sous-classes de graphes (les chaînes et les arbres). Il pourra aussi remettre cette lecture à demain. Notons que la version stochastique en deux étapes du problème de l'arbre couvrant optimal est également étudiée dans cet article.

Une autre approche, plus récente, pour prendre en compte explicitement l'incertitude consiste à munir le problème d'optimisation étudié de *plusieurs jeux de paramètres*, éventuel-

4. Un problème de transport est un problème de flot maximum à coût minimum dans un graphe biparti, le premier sous-ensemble de sommets figurant des sources où sont entreposées des marchandises, et le deuxième sous-ensemble des destinations où s'exercent des demandes en marchandises.

lement probabilisés. Par exemple, si on considère un problème de couplage maximum, on peut envisager k scénarios et valuer chaque arête e du graphe par un vecteur $(w_1(e), \dots, w_k(e))$, où $w_i(e)$ représente le poids de l'arête e si le scénario i se réalise. Il s'agit alors de choisir un couplage sur la base de ses performances dans les différents scénarios. On voit bien à travers cet exemple que la nature d'une solution réalisable n'a pas changé : c'est son évaluation qui est plus fine. En l'absence de probabilités, on parle souvent d'*optimisation combinatoire robuste* (Kouvelis et Yu, 1997). Nos contributions sur cette approche visent, à l'image de notre démarche en optimisation combinatoire multi-objectifs, à autoriser l'emploi de fonctions d'agrégation moins conservatrices que le minimax ou le minimax regret, qui font l'objet de la majorité des travaux dans ce domaine. En particulier, nous nous sommes intéressés aux problèmes d'arbre couvrant robuste et de cheminement robuste dans le cas où les instances sont munies de plusieurs jeux de valuations, que ce soit pour la recherche des solutions Lorenz-optimales ou des solutions OWA-optimales. Nous ne revenons pas dessus dans cette section car ces contributions ont déjà été exposées dans la section précédente, ces problèmes se réduisant à des problèmes multi-objectifs où les objectifs représentent les différents scénarios, et sont donc parfaitement commensurables. Par ailleurs, nous avons étudié le cas où l'on connaît les probabilités d'occurrence des différents scénarios, et où l'on recherche une solution adaptée pour un décideur qui a une aversion pour le risque. Dans un cadre plus conventionnel, nous avons également étudié le modèle d'optimisation combinatoire robuste où les valuations sont de nature intervalle (plutôt que scalaire), et où l'on s'intéresse à la recherche d'une solution minimisant le regret maximum. Ces deux dernières contributions sont détaillées dans la section 5.1.

Enfin, certains formalismes ont été pensés dès leur conception dans le but d'établir des stratégies d'action en environnement incertain. Il s'agit principalement des modèles graphiques tels que les arbres décision hasard (e.g., Raiffa, 1968), les diagrammes d'influence (e.g. Howard et Matheson, 1984), ou les processus décisionnels Markoviens (e.g., Puterman, 1994). Ces modèles permettent de faire de la planification dans l'incertain, aussi appelée parfois *décision séquentielle dans l'incertain* (la première expression est plutôt usitée dans la communauté IA, tandis que la seconde expression est employée par les chercheurs opérationnels et les économistes). Dans ces différents modèles, la détermination d'une stratégie préférée est un problème combinatoire qui a été largement étudié dans la littérature. La plupart des (très nombreux) travaux sur le sujet se fondent sur l'optimisation de l'espérance mathématique pour guider le choix d'une solution. Algorithmiquement, on peut alors avoir recours à la *programmation dynamique stochastique* (à ne pas confondre avec la programmation stochastique) pour calculer une solution optimale. Toutefois, si l'on veut avoir recours à d'autres modèles de décision que l'espérance d'utilité, ou si l'on introduit de l'ambiguïté (c'est-à-dire qu'on relâche l'hypothèse de connaissance parfaite des probabilités), la programmation dynamique stochastique ne s'applique plus et il est nécessaire de proposer de nouvelles méthodes de résolution. C'est la thématique principale de nos travaux en décision séquentielle dans l'incertain, que nous décrivons en section 5.2.

5.1 Optimisation combinatoire robuste ou dans le risque

On distingue deux approches en optimisation combinatoire robuste selon la façon dont l'ensemble des scénarios est défini : le *modèle avec données intervalles* où l'on se donne un intervalle de valeurs possibles pour chaque valuation et où l'ensemble des scénarios est défini en compréhension comme le produit cartésien des intervalles ; le *modèle à scénarios discrets* où chaque valuation est un vecteur (c'est-à-dire qu'on est dans un cadre formel similaire à celui de l'optimisation multi-objectifs) dont chaque composante correspond à un scénario.

Intuitivement, une solution est dite *robuste* si elle reste acceptable quel que soit le scénario qui se réalise. Nous avons travaillé sur chacun des deux modèles mentionnés. Les travaux sur le modèle avec scénarios discrets ont été en partie présentés dans la section dédiée à l’optimisation multi-objectifs. Nous compléterons le panorama en fin de section en indiquant nos contributions dans le cas où les scénarios sont probabilisés. Au préalable, nous décrivons dans les paragraphes suivants les travaux sur le modèle avec données intervalles.

Dans le modèle avec données intervalles, nous avons cherché à identifier des instances polynomiales du plus court chemin robuste et de l’arbre couvrant robuste (Escoffier et al., 2008b,c), problèmes qui ont été prouvés fortement NP-difficiles dans le cas général (Averbakh et Lebedev, 2004). Dans ce but, comme suggéré par Guo et al. (2004), nous avons introduit des paramètres mesurant la distance à des « instances triviales ». L’adjectif “trivial” s’entend ici parfois dans un sens un peu abusif : dans certains cas, une instance est dite triviale car elle peut être résolue en temps polynomial. Par exemple, si tous les intervalles d’une instance se réduisent à un point unique –intervalles *dégénérés*–, alors le problème d’optimisation robuste se réduit à un problème d’optimisation standard, et est par conséquent résoluble en temps polynomial dès lors que la version standard est polynomiale. On peut définir la distance à ce cas trivial comme le nombre k d’intervalles non dégénérés. Si cette distance k est bornée par une constante, alors le problème d’optimisation robuste est résoluble en temps polynomial par un algorithme de force brute (Averbakh et Lebedev, 2004), en testant les 2^k scénarios où chaque intervalle non dégénéré prend comme valeur une de ses extrémités. La complexité paramétrée (Downey et Fellows, 1999) est un cadre formel bien adapté pour ce type d’études. On s’y intéresse en effet à la complexité de résolution d’un problème en fonction de la taille des données d’entrée, *et d’un paramètre k qu’on suppose fixé*. Un problème de données d’entrée de taille n appartient à la classe XP (resp. FPT) si il est résoluble en $O(n^{f(k)})$ (resp. $O(f(k)n^c)$), pour une fonction f quelconque et une constante c quelconque. La classe FPT est un sous-ensemble strict de la classe XP. C’est la classe de complexité la plus avantageuse, car le paramètre k n’y intervient pas en exposant de la taille de l’instance (mais k peut quand même figurer en exposant d’une constante). L’étude d’un problème sous l’angle de la complexité paramétrée permet d’identifier assez précisément ce qui fait la difficulté d’une instance. Dans le travail que nous avons mené dans cette lignée, nous nous sommes focalisés sur deux types de paramètres : ceux qui mesurent la distance à des structures de valuation particulières (instances dont le regret minmax est zéro, instances dont les valuations sont linéairement ordonnées), et ceux qui mesurent la distance à des structures de graphes particulières (graphes série-parallèles, arbres). Nous avons montré que les problèmes de plus court chemin robuste et de couplage robuste dans un graphe biparti sont dans la classe XP, et que le problème de l’arbre couvrant robuste est dans la classe FPT.

À titre d’illustration, considérons le problème de l’arbre couvrant robuste, avec comme paramètre k le nombre d’intervalles qui intersectent au moins un autre intervalle, et montrons que ce problème ainsi paramétré appartient à la classe FPT (Escoffier et al., 2008b,c). Rappelons que, dans le cas général, et même si tous les intervalles sont $[0, 1]$, la détermination d’un arbre couvrant minmax regret est un problème NP-difficile (Aron et Van Hentenryck, 2004; Averbakh et Lebedev, 2004). Pour un nombre k fixé d’intersections d’intervalles, le problème devient polynomial. Ce paramètre semble naturel une fois observé que le cas $k = 0$ (c’est-à-dire que toutes les valuations intervalles sont disjointes) est trivial : dans un tel cas, tout arbre couvrant optimal dans n’importe quel scénario est un arbre couvrant minimisant le regret maximal (Aron et Van Hentenryck, 2004). En effet, l’algorithme de Kruskal mène au même arbre couvrant, quel que soit le scénario retenu, et cet arbre est de regret maximal 0. Supposons maintenant que $k \neq 0$ est fixé. Soit $F \subseteq E$ l’ensemble des arêtes du graphe $G = (V, E)$ dont la valuation intervalle intersecte au moins un autre intervalle, ensemble supposé de cardinal k . Par exemple, sur le graphe de gauche de la figure 14, l’ensemble F est

$\{[a, b], [a, d]\}$ (on a donc $k = |F| = 2$). Soit $F' \subseteq F$. Clairement, si l'on est capable de calculer le meilleur (en terme de regret) arbre couvrant T tel que $T \cap F = F'$ –notons $T_{F'}$ cet arbre–, il suffit de considérer tous les sous-ensembles possibles $F' \subseteq F$, et de retenir le meilleur arbre ainsi calculé. Pour calculer l'arbre $T_{F'}$, on peut procéder comme suit : si F' comporte un cycle, alors il n'existe pas d'arbre contenant toutes les arêtes de F' ; sinon, on supprime $F \setminus F'$ de l'ensemble des arêtes disponibles et, en considérant F' comme des arêtes de $T_{F'}$, on complète cet arbre en appliquant l'algorithme de Kruskal au graphe restant (en choisissant n'importe quelles valuations scalaires compatibles avec les valuations intervalles, puisque l'arbre retourné sera le même quelles que soient les valuations scalaires choisies). Bien que cet algorithme fasse appel à la force brute, la preuve de sa validité, qui consiste à montrer que chaque arbre $T_{F'}$ est bien le meilleur parmi les arbres comportant toutes les arêtes de F' et aucune arête de $F \setminus F'$, n'est pas complètement immédiate (contrairement à ce qu'on pourrait penser) : elle est détaillée dans Escoffier et al. (2008b,c). Comme l'algorithme de Kruskal est de complexité $O(m \log m)$, cet algorithme est de complexité $O(2^k m \log m)$ (puisque'il y a 2^k sous-ensembles F' possibles). Son fonctionnement est illustré sur la figure 14.

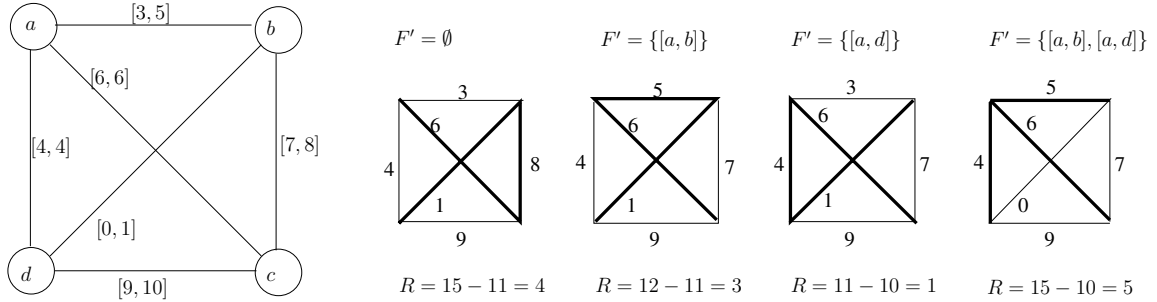


FIGURE 14 – Illustration de l'algorithme de calcul d'un arbre couvrant minimax regret pour un nombre k fixé d'intersections d'intervalles. L'ensemble F est $\{[a, b], [a, d]\}$. Pour chaque sous-ensemble $F' \subseteq F$, on indique en gras l'arbre couvrant optimal parmi ceux qui comportent toutes les arêtes de F' et aucune arête de $F \setminus F'$. Pour calculer le regret maximal pour un arbre T , il faut considérer le pire scénario relatif à cet arbre, c'est-à-dire celui où toutes les valuations sont à leur maximum pour les arêtes de T , et à leur minimum pour les autres arêtes. Le regret maximal est alors la différence, dans cette instanciation, entre la valeur de T et la valeur d'un arbre couvrant minimum. Sur les graphes, les valuations scalaires sont celles du pire scénario relatif à l'arbre en gras, qui permettent de calculer le regret maximal R pour chacun des quatre arbres. L'arbre minimisant le regret maximal est donc $\{[a, c], [a, d], [b, d]\}$.

Dans le modèle à scénarios discrets, nous nous sommes intéressé au cas où l'on connaît les probabilités d'occurrence des différents scénarios (Perny et al., 2007), passant ainsi du cadre de l'incertain total au cadre du risque. Dans ce nouveau cadre, la notion de robustesse s'étend naturellement : il s'agit de trouver une solution optimisant un critère de décision modélisant un comportement adversaire du risque. Un critère bien adapté à ce titre est l'espérance d'utilité (plus précisément de disutilité, car la décision se fonde sur les coûts des solutions potentielles dans les différents scénarios) dépendant du rang (RDW, *Rank Dependent Weight*), qui autorise à modéliser finement l'attitude d'un décideur vis-à-vis du risque, en distinguant un décideur adversaire du risque au sens faible d'un autre adversaire au sens fort (voir section), ce qu'on ne peut faire avec l'espérance d'utilité. Le champs d'application présenté dans l'article susmentionné est la planification d'itinéraire en environnement incertain (incertitude sur l'état du trafic). D'un point de vue algorithmique, le problème peut se voir comme la recherche d'une solution de compromis dans un problème de plus court chemin multi-objectifs,

avec RDW comme fonction d'agrégation. Ainsi sur la figure 15, les arcs sont bi-valués, la première (resp. seconde) composante représentant le coût dans le premier (resp. second) scénario. L'image des différents chemins dans cet espace bi-objectifs (appelé espace des scénarios) est donnée sur le graphique central. Nous avons proposé une approche fondée sur le rangement pour déterminer un itinéraire optimisant RDW pour un décideur adverse du risque, approche qui consiste à énumérer les chemins $x^1, \dots, x^k$ par ordre croissant d'espérance de coût $E(p, y^k)$, où p le vecteur des probabilités et y^k l'image de x^k dans l'espace des scénarios, et à stopper l'énumération lorsque $w(E(p, y^k)) \geq RDW(p, y^*)$, où w la fonction de disutilité dans RDW (jouant un rôle symétrique à la fonction d'utilité u dans RDU) et $RDW(p, y^*) = \min_{i=1, \dots, k} RDW(p, y^i)$. En effet, on exploite la propriété suivante (ainsi que la croissance de w), vérifiée pour un décideur adverse du risque (au sens faible ou au sens fort) : $RDW(p, y) \geq w(E(p, y))$. Lorsque les paramètres de RDW (i.e., la fonction w et la fonction de déformation des probabilités) ne peuvent être élicités, on recherche l'ensemble NSSD (*Non-dominated Second-order Stochastic Dominance*) des solutions stochastiquement non-dominées au second ordre. Pas soucis de concision, nous ne donnons pas ici la définition de la dominance stochastique au second ordre, mais nous préférons donner son interprétation en termes de RDW : (p, y) domine stochastiquement (p, y') au second ordre si et seulement si $RDW(p, y) \leq RDW(p, y')$ pour toute paramétrisation de RDW compatible avec un décideur adverse du risque. Pour rechercher l'ensemble NSSD, nous avons proposé de raffiner un algorithme d'énumération des chemins Pareto-optimaux, car il est facile de prouver que l'ensemble NSSD est inclus dans l'ensemble des solutions Pareto-optimales. Un exemple illustratif est donné sur la figure 15. Les expérimentations numériques menées sur des instances générées aléatoirement ont permis de constater que le cardinal de l'ensemble NSSD restait assez faible (au plus 15), et de résoudre des instances comportant jusqu'à 6000 nœuds et 10 scénarios dans des temps moyens toujours inférieurs à 10 secondes pour la recherche d'une stratégie optimisant RDW, et toujours inférieurs à 30 secondes pour la recherche de l'ensemble NSSD.

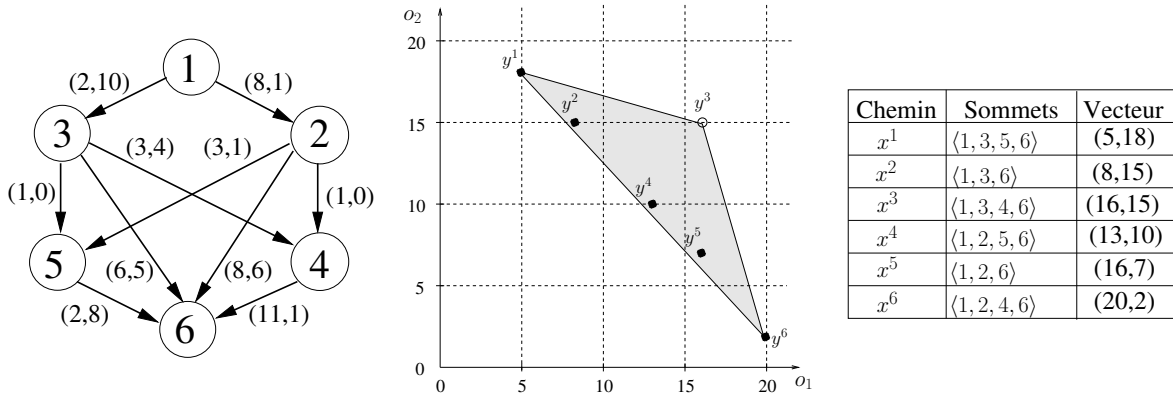


FIGURE 15 – Un problème à deux scénarios et sa représentation. L'ensemble des chemins Pareto-optimaux de 1 à 6 sont $\{x^1, x^2, x^4, x^5, x^6\}$ et, si l'on suppose $p_1 = 0.4$ et $p_2 = 0.6$, l'ensemble NSSD est $\{X^4, X^5, X^6\}$ (ce qui est conforme à l'intuition puisque le premier scénario est moins probable que le second, et que les coûts des chemins x^1 et x^2 sont déséquilibrés avec un pari sur le premier scénario).

Un autre cadre formel bien connu mêlant décision dans l'incertain et aspects combinatoires, et dans lequel nous avons également étudié (entre autres) l'optimisation de RDU, est la décision séquentielle dans l'incertain. Nos travaux sur le sujet sont multiples et font l'objet de la prochaine section, qui conclut la présentation de nos contributions.

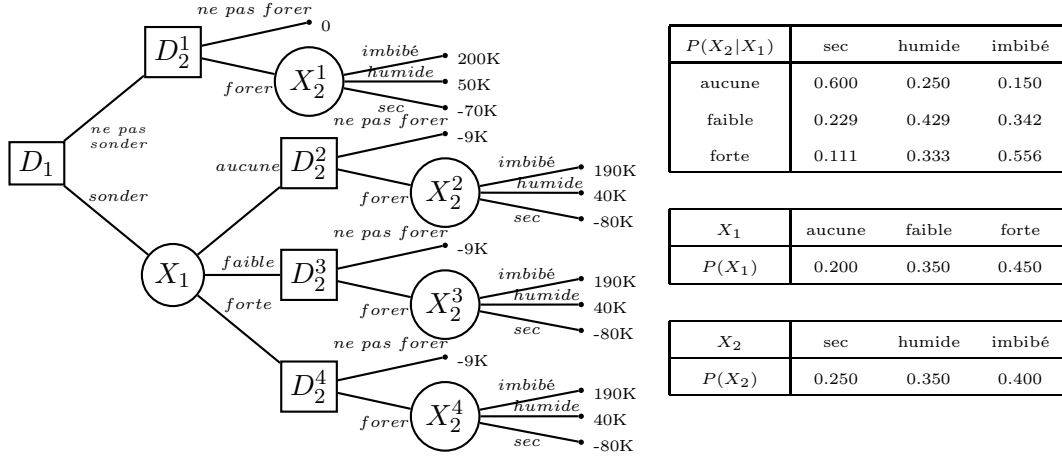


FIGURE 16 – Arbre de décision pour le problème du forage pétrolier.

5.2 Décision séquentielle dans l'incertain

L'un des formalismes les plus populaires pour la planification dans l'incertain est celui des arbres de décision hasard. Un arbre décision hasard fournit une représentation simple et explicite d'un problème de décision séquentielle dans le risque. Il s'agit d'un arbre comportant trois types de nœuds : des nœuds de décision (représentés par des carrés), des nœuds de hasard (représentés par des cercles) et des nœuds d'utilité (les feuilles de l'arbre). A chaque nœud de décision (resp. nœud de hasard) est associée une variable de décision (resp. une variable aléatoire), dont le domaine correspond aux étiquettes des branches issues de ce nœud. Les valeurs indiquées aux feuilles correspondent aux utilités des conséquences. Pour illustration, nous donnons ci-dessous un exemple de problème de décision séquentielle bien connu, et sa représentation sous forme d'un arbre décision hasard. Remarquons qu'on omet les orientations des arêtes quand on représente un arbre de décision.

Exemple 11 (problème du forage pétrolier (Raiffa, 1968)) Une compagnie pétrolière doit décider de forer ou non un site donné. Dans ce but, elle doit d'abord décider si elle sonde ou non la structure géologique du site (décision D_1), ce qui coûte 9000€ et donne une meilleure estimation de la quantité de pétrole qui s'y trouve. Le résultat du sondage peut être vu comme une variable aléatoire X_1 qui peut prendre trois valeurs possibles : aucune si la sonde ne trouve aucune trace de pétrole, faible si la sonde trouve une faible trace de pétrole, ou forte si la sonde trouve une forte trace de pétrole. Ensuite, la compagnie doit décider si elle réalise un forage ou non (décision D_2), ce qui coûte 70000€. Finalement, si la compagnie décide de forer, le résultat du forage peut être vu comme une variable aléatoire X_2 (associée aux nœuds X_2^i pour $i = 1 \dots 4$) qui peut prendre trois valeurs possibles : le puit est sec (le revenu est nul), humide (120000€) ou imbibé (270000€). Ce problème peut être représenté par l'arbre décision hasard sur la gauche de la figure 16, en prenant $u(x) = x$ pour simplifier. Remarquons que la décision D_2 est dupliquée en plusieurs nœuds (nœuds D_2^1 , D_2^2 , D_2^3 et D_2^4) car elle peut être prise dans différents contextes (un sondage a été réalisé ou non, le résultat du sondage est encourageant ou non...). Les probabilités (conditionnelles) d'occurrence des différents événements sont indiquées dans les tables à droite de l'arbre. Ainsi, on a $P(X_2^2 = \text{imbibé}) = P(X_2 = \text{imbibé} \mid X_1 = \text{aucune}) = 0.150$.

Une stratégie consiste à fixer une valeur à chaque variable de décision conditionnellement à son passé. Par exemple, l'arbre de décision de la figure 16 comporte 10 stratégies réalisables, parmi lesquelles figure, entre autres, la stratégie $s = \{D_1 = \text{sonder}, D_2^2 = \text{forer}, D_2^3 =$

$forer, D_2^4 = forer\}$ (remarquons que le nœud D_2^1 ne peut pas être atteint quand $D_1 = sonder$). Dans un arbre décision hasard binaire complet, le nombre de stratégies est en $\Theta(2^{\sqrt{n}})$, où n le nombre de nœuds de décision figurant dans l'arbre. Ainsi, le nombre de stratégies potentielles est exponentiel dans la taille de l'instance, et il est par conséquent nécessaire de développer un algorithme d'optimisation combinatoire pour déterminer la stratégie optimale. Pour le modèle EU, il est bien connu qu'il existe un algorithme linéaire (i.e., en $O(n)$) permettant de déterminer, par programmation dynamique, une stratégie optimale. En effet, une telle stratégie vérifie le principe d'optimalité : toute sous-stratégie d'une stratégie optimale est optimale. L'idée de l'algorithme consiste donc à procéder par induction arrière à partir des nœuds terminaux, afin de déterminer en chaque nœud l'espérance d'utilité d'une sous-stratégie optimale :

- en un nœud de hasard, l'espérance d'utilité optimale est égale à l'espérance des utilités optimales de ses successeurs ;
- en un nœud de décision, l'espérance d'utilité optimale est égale à la plus grande des espérances d'utilité optimales de ses successeurs.

Par exemple, dans l'arbre décision hasard de la figure 16, en X_2^2 l'espérance de gain est de $-9.5K$: on a donc intérêt à ne pas forer en D_2^2 , et l'espérance de gain optimale en D_2^2 est de $-9K$. En remontant ainsi l'arbre des feuilles vers la racine, on obtient l'espérance de gain optimale en D_1 .

Néanmoins, ce cadre habituel n'est pas toujours adapté. D'une part, on peut vouloir faire appel à une autre modèle de décision que l'espérance d'utilité (tel que l'espérance d'utilité dépendant du rang par exemple). D'autre part, on peut ne pas disposer d'une information parfaite sur les probabilités ou sur les conséquences. Dans un cas comme dans l'autre, l'approche par programmation dynamique n'est la plupart du temps plus valide, car le principe d'optimalité n'est plus vérifié : une stratégie optimale est susceptible de comporter des sous-stratégies sous-optimales. Deux directions de recherche peuvent alors être adoptées. La première, suivie par Kikuti et al. (2011), est d'adopter une hypothèse de *faisabilité dynamique* (Seidenfeld, 1988), qui consiste à supposer que le décideur anticipe ses décisions futures et écarte donc d'emblée toutes les stratégies comportant des sous-stratégies sous-optimales (car il anticipe qu'il souhaitera en dévier). Cela revient à rechercher la stratégie retournée par induction arrière dans l'arbre décision hasard (combien même le modèle de décision adopté ne vérifie pas le principe d'optimalité). Cette stratégie sera préférée par un décideur *conséquentialiste*, c'est-à-dire dont les décisions ne dépendent ni du passé, ni d'événements contrefactuels (autrement dit, les événements qui auraient pu se réaliser). Bien que séduisante d'un point de vue algorithmique, cette approche peut conduire le décideur à adopter une stratégie "dominée" en un certain sens (par exemple, la stratégie calculée par induction arrière avec l'espérance d'utilité dépendant du rang peut être stochastiquement dominée). La seconde direction de recherche, proposée par McClennen (1990), est de remettre en cause le conséquentialisme et d'adopter une hypothèse de *choix résolu*, qui consiste à supposer que le décideur se fixe une stratégie initialement, et ne s'autorise pas à en dévier par la suite. Il s'agit donc d'adopter un comportement inverse de celui que décrivait Sacha Guitry dans ses *Pensées, maximes et anecdotes* :

« Mesdames, il nous est difficile de revenir aussi vite que vous sur les décisions que vous prenez. » ; -)

En considérant chaque nœud de décision comme un *ego* indépendant des autres (différentes "incarnations" d'un même décideur dans plusieurs contextes décisionnels distincts), on peut distinguer en particulier deux manières de faire du choix résolu :

- *ego dictateur à la racine* : déterminer une stratégie optimale du point de vue de l'ego à la racine et appliquer ensuite la stratégie sans jamais en dévier ;
- *compromis entre les egos* : déterminer une stratégie qui satisfait équitablement tous les

egos présents dans la stratégie (c'est-à-dire dont chaque sous-stratégie est « proche » d'être optimale).

Dans un cas comme dans l'autre, cela soulève des problèmes algorithmiques intéressants, de par le nombre combinatoire de stratégies potentielles. Dans la suite de la section, nous rapportons nos travaux algorithmiques sur le choix résolu. Pour des raisons de place, nous nous focalisons exclusivement sur le cas de l'ego dictateur à la racine (bien que nous ayons également étudié la problématique de recherche d'un compromis entre les egos). Nous nous sommes principalement intéressés à deux modifications du cadre habituel nécessitant l'emploi du choix résolu : l'optimisation de RDU dans les arbres décision hasard (les probabilités étant alors supposées connues avec précision) et l'optimisation de Γ -maximin lorsque les probabilités sont mal connues.

Optimisation de RDU dans les arbres décision hasard. Le développement de procédures opérationnelles de choix résolu est un prérequis à l'emploi de RDU dans des problèmes de décision séquentielle. En réponse à ce besoin, et dans le cadre d'une approche de type « compromis entre les egos », Jaffray et Nielsen (2006) ont proposé un algorithme permettant de retourner une stratégie dont toutes les sous-stratégies sont à la fois proches de l'optimum pour RDU et stochastiquement non-dominées (sous réserve qu'une telle stratégie existe).

Notre propre contribution à l'utilisation de RDU dans les arbres décision hasard a consisté dans le développement d'algorithmes exacts pour déterminer une stratégie optimale du point de vue de l'ego dictateur à la racine (Jeantet et Spanjaard, 2008, 2011). Nous avons tout d'abord montré que le problème est NP-difficile en établissant une réduction depuis 3-SAT (e.g., Garey et Johnson, 1979). Afin de ne pas entrer dans des détails techniques, nous nous bornons ici à donner une intuition de la preuve. Etant donnée une formule 3-SAT, il s'agit de générer en temps polynomial un arbre décision hasard et une fonction RDU en sorte que la détermination d'une stratégie optimale permette de conclure quant à la satisfiabilité ou non de la formule. Nous allons donner l'intuition via un exemple. Considérons la formule $(x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_3 \vee x_4) \wedge (\overline{x_2} \vee \overline{x_3} \vee \overline{x_4})$. L'arbre décision hasard généré est représenté sur la figure 17. Pour chaque variable x_i de la formule, on a donc créé un nœud de décision avec deux fils (nœuds de hasard), le nœud V_i (resp. F_i) correspondant à l'affectation de la valeur de vérité *vrai* (resp. *faux*). À une stratégie correspond ainsi clairement une affectation de valeurs de vérité aux variables de la formule. Les conséquences auxquelles est susceptible de mener un nœud de hasard correspondent aux différentes clauses c_i rendues vraies par l'affectation correspondante, où $c_1 = x_1 \vee x_2 \vee x_3$, $c_2 = \overline{x_1} \vee x_3 \vee x_4$ et $c_3 = \overline{x_2} \vee \overline{x_3} \vee \overline{x_4}$ (la clause fictive c_0 est systématiquement présente, pour des raisons techniques que nous ne détaillerons pas ici). De ce fait, la formule est satisfiable si et seulement si il existe une stratégie dans l'arbre pour laquelle l'ensemble des clauses figurent parmi les conséquences possibles. Par exemple, la formule du dessus est satisfiable car la stratégie $\{x_1=\text{vrai}, x_2=\text{faux}, x_3=\text{vrai}, x_4=\text{vrai}\}$ peut conduire aux conséquences c_1 , c_2 ou c_3 . Pour terminer la réduction, il faut encore fixer les probabilités de l'arbre, et paramétrer la fonction RDU de manière idoine afin de détecter si l'ensemble des clauses figurent parmi les conséquences possibles. C'est bien évidemment la présence de la fonction de déformation des probabilités qui rend cela possible. Le lecteur intéressé pourra trouver la preuve détaillée dans Jeantet et Spanjaard (2008).

Afin de déterminer une stratégie optimale vue de la racine selon RDU, nous avons proposé une procédure de branch and bound dont les principales caractéristiques sont les suivantes :

- le principe de branchement consiste simplement à instancier une variable de décision ;
- l'obtention d'une borne supérieure en un nœud du *branch and bound* se fait en calculant par induction arrière une loterie qui domine stochastiquement toutes les stratégies compatibles avec les instanciations caractérisant ce nœud, puis en évaluant cette loterie

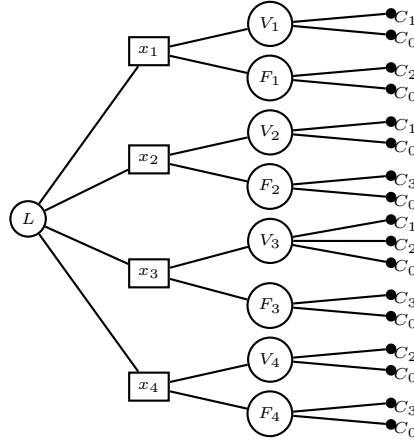


FIGURE 17 – La détermination d’une stratégie RDU-optimale est un problème NP-difficile.

selon RDU (on s’assure ainsi de borner supérieurement de par la compatibilité de RDU avec la dominance stochastique) ;

- l’obtention d’une borne inférieure se fait en calculant une stratégie optimale selon EU parmi celles compatibles avec les instanciations caractérisant le nœud, puis en évaluant cette stratégie selon RDU ;

Cet algorithme permet de résoudre des arbres décision hasard binaires complets comportant jusqu’à 32000 nœuds en un temps moyen inférieur à la minute (sur une machine équipée d’un processeur Pentium IV CPU 2.3Ghz et de 3.5Go de RAM), quelle que soit la fonction de déformation des probabilités utilisée. Ces performances sont tout-à-fait satisfaisantes dans la mesure où de plus grandes instances deviennent difficilement stockables en mémoire sur un ordinateur personnel standard actuel. Une formulation MIP a également été proposée lorsque la fonction φ de déformation est concave et linéaire par morceaux (Jeantet et Spanjaard, 2011), qui présente l’avantage de la simplicité d’implantation (en faisant appel à un solveur) mais dont les performances sont moindres puisqu’un temps de résolution de l’ordre de la minute (sur la même machine) est atteint dès 8000 nœuds. Toutefois, il est notable que les arbres décision hasard générés dans ces tests sont particulièrement denses dans la mesure où ce sont des arbres parfaitement équilibrés. En pratique, dans de nombreuses applications, les arbres décision hasard sont déséquilibrés et on peut par conséquent penser que, pour le même nombre de nœuds de décision, une stratégie optimale sera calculée plus vite. Pour s’en convaincre, des expérimentations numériques ont été menées sur une modélisation du célèbre jeu télévisé « Qui veut gagner des millions ? »⁵. Cette modélisation, qui enrichit un modèle proposé par Perea et Puerto (2007) (qui ne tenait pas compte de la connaissance ou non de la réponse à la question posée dans le choix d’utiliser un joker), comporte plus de 75 millions de nœuds. La génération de l’instance requérant une mémoire vive importante, la résolution par *branch and bound* a été mise en œuvre sur un cluster de machines disposant de 64Go de RAM. Le temps de résolution est de l’ordre de l’heure, que la fonction φ soit concave ou convexe (avec néanmoins un temps de résolution plus important dans le cas d’une fonction φ convexe, ce qui correspond à un goût pour le risque si on suppose la fonction d’utilité linéaire).

5. Dans ce jeu, un candidat est confronté à une série de 15 questions, chaque nouvelle réponse correcte augmentant ses gains de façon exponentielle. A chaque question, le candidat peut décider de répondre, au risque de perdre une grande partie des gains cumulés jusqu’alors, ou de partir avec le montant actuel de ses gains. Le candidat dispose de trois jokers pour toute la partie, auxquels il peut faire appel au moment où il le souhaite.

Comme le montre bien l'exemple de « Qui veut gagner des millions ? », ces approches se heurtent au principal inconvénient des arbres décision hasard : leur taille peut rapidement devenir prohibitive lorsque le nombre d'étapes de décision croît. Pour cette raison, nous avons également étudié l'optimisation de RDU dans les *diagrammes d'influence* (Howard et Matheson, 1984; Shachter, 1986), qui sont des modèles graphiques qui permettent une représentation compacte des arbres décision hasard. Les diagrammes d'influence tirent en effet parti de la décomposabilité de la structure probabiliste des problèmes (plus précisément, ils exploitent les indépendances probabilistes du problème pour factoriser certaines données). La détermination d'une stratégie EU-optimale dans un diagramme d'influence est déjà un problème NP-difficile. Néanmoins, des algorithmes performants utilisant des techniques importées des réseaux bayésiens existent pour le résoudre lorsque l'on cherche à optimiser EU (Jensen et al., 1994). Comme pour les arbres décision hasard, ces algorithmes ne s'étendent pas pour l'optimisation du critère RDU du fait que la programmation dynamique n'est plus valide. C'est pourquoi nous avons développé un branch and bound permettant de « résoudre » un diagramme d'influence au sens de RDU (Jeantet et Spanjaard, 2009, 2011). Les principales caractéristiques de cet algorithme sont assez similaires à celles du branch and bound dédié aux arbres décision hasard, à la différence que l'on doit faire appel à de la programmation dynamique non-sérielle (Bertelè et Brioschi, 1972) pour calculer une loterie qui domine stochastiquement toutes les stratégies du diagramme, là où on employait auparavant de la programmation dynamique simple. L'algorithme développé ne garantit toutefois pas de retourner la stratégie optimale au sens de RDU. En effet, la différence la plus significative entre le cas des arbres décision hasard et celui des diagrammes d'influence vient de ce que l'ensemble des stratégies potentielles envisagées dans un diagramme d'influence est un sous-ensemble strict de l'ensemble des stratégies que l'on obtiendrait si on « dépliait » le diagramme pour générer l'arbre décision hasard correspondant. Si, dans le cas de EU, cela ne porte pas à conséquence car on est assuré qu'il existe une stratégie EU-optimale parmi celles envisagées, il n'en est pas de même dans le cas de RDU : il se peut en effet que toutes les stratégies RDU-optimales soient extérieures au sous-ensemble de stratégies envisagées. Une voie intermédiaire possible est alors de modifier le diagramme d'influence afin d'élargir le sous-ensemble de stratégies envisagées. On réalise ainsi un compromis entre compacité de la représentation et qualité de la solution retournée au sens de RDU. C'est la voie que nous avons explorée, avec des résultats proches de l'optimum (Jeantet et Spanjaard, 2009, 2011).

Optimisation de Γ -maximin dans les arbres décision hasard. Un autre contexte qui confronte à des difficultés algorithmiques nouvelles est celui des arbres décision hasard où les probabilités sont mal connues. En effet, lorsque plusieurs experts ont des points de vue divergents ou quand certaines données empiriques manquent, il n'est pas évident d'élucider des probabilités précises pour chacun des événements. Une manière naturelle de prendre en compte explicitement cette difficulté est de définir un jeu de probabilités compatibles avec les informations dont on dispose, soit en valuant les branches issues des nœuds de hasard par des intervalles de probabilités, soit en définissant des contraintes sur les probabilités en ces nœuds. C'est le cadre des probabilités mal connues, aussi appelé ambiguïté, décrit dans la section 3.2.3. Pour guider la recherche d'une stratégie dans ce cadre, nous nous sommes focalisés, comme pour RDU, sur la recherche d'une stratégie optimale vue de la racine pour un critère donné de décision dans l'ambiguïté. On se heurte principalement à deux difficultés lorsque l'on cherche à concevoir des algorithmes la décision séquentielle en présence d'ambiguïté :

- *il n'est pas suffisant de s'assurer que les probabilités conditionnelles somment à 1 en chaque nœud de hasard pour garantir la cohérence globale de la distribution de probabilité sur l'arbre décision hasard.* A titre d'illustration, reprenons l'exemple du forage pétrolier en supposant cette fois-ci que les probabilités sont imprécises et définies selon la table 6.

Les probabilités conditionnelles suivantes sont compatibles avec les intervalles et peuvent être complétées de façon à ce qu'elles somment à 1 en chaque nœud : $P(X_2 = \text{sec}|X_1 = \text{aucune}) = 0.55$, $P(X_2 = \text{sec}|X_1 = \text{faible}) = 0.33$, $P(X_2 = \text{sec}|X_1 = \text{forte}) = 0.12$, $P(X_1 = \text{aucune}) = 0.20$, $P(X_1 = \text{faible}) = 0.35$, $P(X_1 = \text{forte}) = 0.45$, $P(X_2 = \text{sec}) = 0.22$. Pourtant, aucune distribution de probabilité valide ne peut induire de telles probabilités conditionnelles, car la formule des probabilités totales n'est pas vérifiée : $P(X_2 = \text{sec}|X_1 = \text{aucune})P(X_1 = \text{aucune}) + P(X_2 = \text{sec}|X_1 = \text{faible})P(X_1 = \text{faible}) + P(X_2 = \text{sec}|X_1 = \text{forte})P(X_1 = \text{forte}) = 0.2795 \neq 0.22 = P(X_2 = \text{sec})$. Il existe toutefois certains arbres décision hasard où cette difficulté ne survient pas : ces arbres sont dits *séparables* (*separately specified* pour Kikuti et al. (2011)) car il suffit que les probabilités conditionnelles somment à 1 en chaque nœud de hasard pour que la distribution de probabilité soit globalement valide. Par opposition, les autres arbres sont dits *non-séparables*.

$P(X_2 X_1)$	sec	humide	imbibé
aucune	[0.500,0.666]	[0.222,0.272]	[0.125,0.181]
faible	[0.222,0.333]	[0.363,0.444]	[0.250,0.363]
forte	[0.111,0.166]	[0.333,0.363]	[0.454,0.625]

X_1	aucune	faible	forte
$P(X_1)$	[0.181,0.222]	[0.333,0.363]	[0.444,0.454]

X_2	sec	humide	imbibé
$P(X_2)$	[0.214,0.344]	[0.309,0.386]	[0.307,0.456]

TABLE 6 – Probabilités imprécises pour le problème du forage pétrolier.

- pour la plupart des modèles de décision en présence de probabilités mal connues, le principe d'optimalité n'est pas vérifié, autrement dit une stratégie optimale n'est pas nécessairement constituée de sous-stratégies optimales. A titre d'illustration, considérons un jeu où l'on commence par lancer une pièce bien équilibrée, avant de tirer une boule dans une urne –dite d'Ellsberg– comportant une proportion $p_R = 1/3$ de boules rouges et une proportion $p_N + p_J = 2/3$ de boules noires ou jaunes (l'urne ne change pas selon le résultat du lancer). Si la pièce tombe sur pile (resp. face), on demande au joueur de parier sur le rouge ou le noir (resp. le rouge ou le jaune). Dans les deux cas (pile ou face), l'utilité est supposée de $1 - \varepsilon$ si le joueur a parié sur le rouge et que le rouge sort, de 1 si le joueur a parié sur le noir (resp. jaune) et que le noir (resp. jaune) sort, et de 0 sinon. Le jeu peut être représenté par l'arbre décision hasard de la figure 18. Supposons que le joueur suit le modèle Γ -maximin, c'est-à-dire que ses choix sont guidés par la pire espérance d'utilité possible. Remarquons que, une fois le résultat du lancer de pièce connu (que ce résultat soit pile ou face), choisir le rouge conduit à une espérance d'utilité de $(1 - \varepsilon)/3$ au pire, contre 0 pour la couleur alternative (au pire des cas, il n'y a aucune boule de cette couleur dans l'urne). En procédant par induction arrière, la stratégie obtenue consiste donc à parier sur le rouge quel que soit le résultat du lancer de pièce, pour une espérance globale d'utilité de $(1 - \varepsilon)/3$ au pire. Pourtant, en adoptant la stratégie consistant à parier sur le noir si la pièce tombe sur pile, et sur le jaune si la pièce tombe sur face, on se convaincra sans mal que l'espérance globale d'utilité est $\min_{p_N, p_J} (1/2)p_N + (1/2)p_J = \min_{p_N, p_J} (1/2)(p_N + p_J) = 1/3$ au pire, et donc que cette stratégie est meilleure selon le critère Γ -maximin.

En amont de la détermination d'une stratégie optimale, on peut s'intéresser au problème plus simple de l'évaluation d'une stratégie donnée. Lorsque les probabilités sont mal connues, nous avons montré que l'évaluation d'une stratégie est un problème NP-difficile en soi (Fargier et al., 2011), de par les multiples occurrences possibles d'une même variable aléatoire en différents nœuds de l'arbre. Face à ce constat, nous avons toutefois proposé une formulation en programmation mathématique pour réaliser l'évaluation d'une stratégie, formulation qui comporte un nombre de variables exponentiel du nombre de variables aléatoires figurant

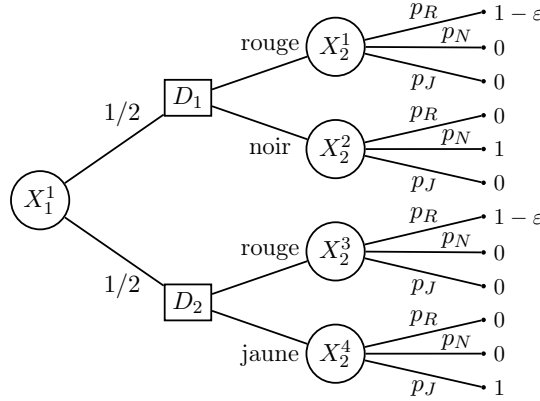


FIGURE 18 – Un arbre décision hasard impliquant une urne d’Ellsberg. La variable X_1 correspond au lancer de la pièce. En D_1 le résultat est pile, et en D_2 le résultat est face. La variable X_2 (présente en plusieurs exemplaires) correspond au tirage dans l’urne.

dans l’arbre. Le travail principal a consisté à définir des variables et des contraintes permettant de caractériser l’ensemble des distributions de probabilité valides sur l’arbre (Jeantet et Spanjaard, 2010) : pour ce faire, nous nous sommes ramenés à la loi de probabilité jointe de l’ensemble des variables aléatoires figurant dans l’arbre, ce qui permet de s’assurer simplement de la cohérence globale de la distribution de probabilité en posant une contrainte spécifiant que les probabilités jointes somment à 1.

Concernant la détermination proprement dite d’une stratégie optimale pour Γ -maximin vue de la racine, nous avons proposé deux approches selon la nature séparable ou non de l’arbre décision hasard (Fargier et al., 2011) :

- dans les arbres décision hasard séparables, on peut procéder récursivement, ce qui requiert la résolution d’un (petit) programme linéaire en chaque nœud de hasard ;
- dans les arbres décision hasard non-séparables, on peut procéder en deux phases, en calculant tout d’abord l’ensemble des stratégies non-dominées (voir section 3.2.3 pour la définition) –ce qui peut se faire par programmation dynamique–, puis en scannant cet ensemble pour obtenir une stratégie optimale au sens de Γ -maximin (il existe nécessairement une stratégie optimale pour Γ -maximin qui est non-dominée).

L’algorithme pour les arbres séparables permet de résoudre des arbres décision hasard binaires complets comportant jusqu’à 2 millions de nœuds en un temps moyen inférieur à la minute (sur une machine équipée d’un processeur PIV CPU 2.3Ghz et de 3.5Go de RAM) : dans les test conduits, les arbres sont munis de probabilités intervalles $[p_i^-, p_i^+]$ avec $p_i^+ - p_i^- = 0.1$. Pour les arbres non-séparables, l’approche en deux phases permet de résoudre des arbres de tailles plus modestes : pour la même structure d’arbre et le même niveau d’imprécision dans les probabilités, si on veut obtenir des temps de résolution qui restent sous la minute, on ne peut monter au-delà de 2000 nœuds et 8 variables aléatoires distinctes.

Ceci clôt la présentation de nos principales contributions en décision séquentielle dans l’incertain. À la manière de notre démarche en optimisation combinatoire multi-objectifs, nous avons cherché à promouvoir l’emploi de modèles de décision sophistiqués (autres que l’espérance d’utilité) en proposant des algorithmes adaptés. Certains des grands schémas algorithmiques employés en optimisation multi-objectifs ont pu être fructueusement importés en décision séquentielle dans l’incertain, en particulier l’idée de calculer un ensemble de solutions non-dominées parmi lesquelles on peut prouver qu’il existe une solution optimale pour le modèle de décision étudié. Cette approche n’a bien sûr de sens que si la règle de dominance employée se prête à la programmation dynamique.

6 Conclusion et directions de recherche futures

6.1 Synthèse

Dans ce mémoire, après avoir explicité la conception de l'aide à la décision dans laquelle nous nous reconnaissons et avoir réalisé une brève présentation de la théorie de la décision algorithmique dans laquelle s'inscrivent nos travaux, nous nous sommes efforcés de synthétiser l'essentiel de nos contributions, avec le choix délibéré de se situer à un niveau de détails suffisant pour permettre au lecteur d'en saisir concrètement les idées clés. Nous espérons y être parvenu, au moins partiellement. Nos travaux ont porté principalement sur l'optimisation combinatoire multi-objectifs et l'optimisation dans l'incertain. Ils partagent le point commun de porter sur des méthodes de résolution exactes ou approchées avec garantie de performance.

En optimisation combinatoire multi-objectifs, une de nos principales contributions est le développement de bornes multi-objectifs à plusieurs points (et même un continuum) pour réaliser une approximation plus fine de la frontière de Pareto, et le développement d'algorithmes d'énumération implicite exploitant ces bornes multi-objectifs. Cela nous a permis en particulier de développer des algorithmes efficaces pour l'arbre couvrant bi-objectifs (par branch and bound multi-objectifs, cadre formel que nous avons contribué à développer) et le sac-à-dos bi-objectifs (par programmation dynamique hybride). De plus, lorsque l'on ne cherche plus à identifier le front de Pareto entier mais seulement une solution de compromis au sein de ce front, nous avons proposé des bornes et développé des procédures de résolution pour optimiser en particulier la moyenne ordonnée pondérée et l'intégrale de Choquet. Focaliser ainsi sur une solution de compromis particulière permet de s'attaquer à des problèmes à plus de trois objectifs, ce qui ne serait pas envisageable avec une approche Pareto.

En optimisation dans l'incertain, nous nous sommes intéressés en particulier à l'optimisation de modèles non-EU en décision séquentielle, parmi lesquels l'utilité espérée dépendant du rang et le critère de Hurwicz pour les probabilités imprécises. L'originalité de nos contributions dans ce domaine se situe selon nous dans la vision algorithmique que nous avons adoptée, qui avait été peu abordée jusqu'alors. Les algorithmes que nous proposons nous semblent combler un manque, car ils constituent un premier pas vers l'utilisation de modèles non-EU dans un contexte séquentiel et opérationnel. Hormis ces travaux en décision dans l'incertain, nous avons également développé une certaine activité en optimisation robuste, que ce soit dans des modèles avec un jeu discret de scénarios (proches de l'optimisation multi-objectifs) ou dans des modèles avec données intervalles. Ici encore, la singularité de nos travaux réside dans le choix des critères de décision à optimiser et dans le souci d'obtenir des méthodes de résolution exactes, comme par exemple lorsque nous nous sommes intéressés à identifier des classes d'instances en optimisation robuste avec données intervalles pour lesquels on peut proposer des algorithmes de complexité polynomiale en la taille de l'instance et exponentielle en un paramètre dont la valeur reste faible.

La partie de nos travaux portant sur l'optimisation approchée avec garantie de performance est plus réduite, mais offre une alternative intéressante aux approches exactes, en permettant d'obtenir des solutions de qualité en des temps très compétitifs. En optimisation multi-objectifs, la notion de frontière de Pareto ε -approchée présente l'avantage de contourner la difficulté inhérente à la détermination exacte de la frontière de Pareto : le nombre pseudo-polynomial de points dans la frontière. En effet, comme nous l'avons rappelé, il existe toujours une frontière ε -approchée comportant un nombre polynomial de points. Nos travaux sur ce sujet se limitent aujourd'hui à un unique article, mais pourraient être étendus. Pour ce faire, il nous semble pertinent d'étudier la détermination de frontières approchées ne comportant que des solutions Pareto-optimales (ce qui n'est pas requis dans la définition standard d'une frontière ε -approchée). Concernant l'optimisation dans l'incertain, les travaux relevant

de l'optimisation approchée avec garantie de performance sont ceux conduits sur l'optimisation combinatoire à deux étapes. Cette problématique nous semble assez naturelle dans la mesure où la possibilité d'adapter sa stratégie après acquisition d'information est assez réaliste.

6.2 Directions de recherche futures

Dans l'avenir, tout en continuant notre activité en optimisation multi-objectifs et dans l'incertain, nous souhaitons élargir notre champs d'intérêt à d'autres thématiques au carrefour de l'économie et de l'algorithmique, telles que la théorie des jeux algorithmique et le choix social computationnel. Nous décrivons maintenant plus précisément les travaux que nous envisageons ou que nous avons déjà initiés dans cette optique.

Optimisation combinatoire multi-objectifs. Dans le cadre du projet ANR GUEPARD (*GUaranteed Efficiency for PAReto optimal solutions Determination*), portant sur l'optimisation combinatoire multi-objectifs, nous souhaitons poursuivre nos travaux sur la détermination du front de Pareto ou la recherche d'une solution de meilleur compromis. Nous donnons ici quelques voies de recherche qui nous semblent intéressantes à explorer. Dans les travaux que nous envisageons, nous distinguons le cas multi-objectifs et le cas « multi-agents ». Nous entendons par optimisation multi-agents des problèmes où le nombre d'objectifs est une variable du problème et non plus une constante fixée à l'avance. C'est le cas par exemple dans le problème d'affectation multi-agents. Il se formule comme un problème d'affectation mono-objectif, à la différence qu'une affectation est caractérisée non plus par la somme des coûts individuels qui la composent, mais par le vecteur des coûts individuels. Dans le cadre de l'équité, on s'intéresse alors à chercher une affectation pour laquelle le profil des disutilités des agents est équilibré. Pour illustrer notre propos, nous reprenons un exemple de problème d'affectation proposé par Golden et Perny (2010), où c_{ij} représente le coût d'affecter la tâche j à l'agent i :

$$C = \begin{pmatrix} 5 & 8 & (4) & 9 & \mathbf{7} \\ \mathbf{1} & (3) & 2 & 7 & 8 \\ (3) & 9 & \mathbf{2} & 9 & 5 \\ 10 & 1 & 3 & (\mathbf{3}) & 4 \\ 5 & \mathbf{1} & 7 & 7 & (3) \end{pmatrix}$$

L'affectation optimale au sens de la somme des coûts est indiquée en gras. Elle coûte 14 et on voit que l'agent 1 est largement lésé puisqu'il subit à lui seul un coût de 7. L'affectation indiquée entre parenthèses vaut 16 mais le profil est beaucoup plus équilibré. C'est précisément ce type d'affectation que l'on cherche à retourner en optimisation multi-agents équitable. Un autre exemple de problème d'optimisation multi-agents est le problème de la localisation d'une ressource partagée sur un réseau (un hôpital dans une ville par exemple), où chaque nœud du réseau représente un agent. Ces problèmes sont fréquemment abordés aujourd'hui sous l'angle de la programmation mathématique, qui fournit un outil algorithmique commode pour traiter un grand nombre d'objectifs sans un ralentissement trop considérable.

Nous envisageons d'aborder ces problèmes avec le même outil algorithmique, mais en s'intéressant à des fonctions d'utilité collectives issue de la théorie de la décision, en lieu et place des fonctions d'utilité *minmax* ou *somme* usuellement adoptées. Le travail consistera donc à proposer des linéarisations efficaces des fonctions d'utilité collectives les mieux établies. Des premiers résultats dans cette direction ont déjà été présentés par Lesca et Perny (2010), qui montrent par exemple comment la propriété de 2-additivité d'une capacité permet de proposer une linéarisation efficace de l'intégrale de Choquet (une capacité est dite 2-*additive* si elle est complètement caractérisée par les capacités des singletons et des paires). Cette piste de recherche est aussi intéressante dans le cas plus purement multi-objectifs, autrement dit quand

le nombre d'objectifs est une constante du problème. Un premier travail sur l'optimisation de la moyenne ordonnée pondérée dans des problèmes d'arbres couvrants multi-objectifs a d'ores et déjà été conduit (Galand et Spanjaard, 2012), avec des performances numériques améliorant celles obtenues par les approches d'énumération implicite dédiées grâce à des procédures de prétraitement efficaces permettant de réduire considérablement la taille des instances. L'étude d'autres problèmes combinatoires multi-objectifs sous cet angle nous semble donc une bonne piste.

Un autre paradigme que la programmation mathématique, nous semblant pouvoir être fructueusement exploité dans le cadre de la recherche d'une solution de meilleur compromis, est celui de la programmation dynamique. En effet, en jouant sur les mêmes conditions et les mêmes arguments que ceux mis en avant dans la section 4.2.4 pour généraliser les règles des cycles et des cocycles dans les problèmes d'arbre couvrant optimal au sens d'un OWA, on peut réaliser une procédure de programmation dynamique pour la détermination d'un meilleur compromis au sens d'une fonction d'agrégation φ , procédure fondée sur la relation $\varphi(y' - y) \leq 0$ (selon le sens d'optimisation). Parallèlement à nos travaux sur le problème de l'arbre couvrant multi-objectifs, Fouchal et al. (2011) ont d'ailleurs proposé une procédure de programmation dynamique fondée sur une "dominance de Choquet", plus riche que la dominance de Pareto, pour le problème de plus court chemin multi-objectifs.

Dans les directions de recherche plus spécifiques à un problème particulier, mentionnons la compilation dans des problèmes de détermination multi-objectifs d'itinéraires. Dans des réseaux où de multiples requêtes d'itinéraires entre deux points seront faites (réseaux de type *Google maps*), il s'agit de « prétraiter » l'instance de façon à ce que les calculs soient ensuite accélérés lors des réponses aux requêtes. Les informations retournées par ces calculs doivent de plus être stockables de façon compacte. De nombreux travaux sur ce sujet ont été menés pour la version mono-objectif du problème (voir par exemple Goldberg, 2007). Nous avons commencé à étudier ce qui peut être fait pour la version multi-objectifs.

Enfin, nous envisageons de développer une activité dans la mise en œuvre de l'optimisation multi-objectifs dans des contextes applicatifs. En particulier, nous avons initié un travail sur l'optimisation de la qualité de service dans les services webs composites, et plus précisément sur la recherche de compositions équitables entre utilisateurs (El Haddad et Spanjaard, 2009).

Théorie des jeux algorithmique. Dans le cadre du projet ANR COCA (*Combinatorial Optimization with Competing Agents*), portant sur la théorie des jeux algorithmique (Nisan et al., 2007), nous nous intéressons à des jeux où l'espace des stratégies potentielles (pour chaque joueur) est combinatoire, et étudions l'impact que peut avoir une entité extérieure qui essaie de conduire le système vers un équilibre socialement performant (en mettant en place des mécanismes incitatifs). Rappelons qu'un équilibre de Nash est une situation devenue stable, caractérisée par le fait que tout joueur déviant unilatéralement de sa stratégie affaiblit sa position personnelle (la figure 19 représente un exemple didactique d'équilibre de Nash, proposé par Corinne Touati). Pour illustration, nous reprenons ici un célèbre exemple dû à Pigou (1920), économiste britannique bien connu. Considérons une population d'agents souhaitant se rendre d'une même origine à une même destination sur un réseau routier, avec un choix entre deux itinéraires alternatifs. Le premier itinéraire nécessite une heure, quel que soit le nombre d'utilisateurs l'empruntant. Le temps de parcours pour le second itinéraire est par contre proportionnel au nombre d'utilisateurs l'empruntant. Par exemple, si 50% de la population emprunte cet itinéraire, le temps de parcours est de 30 minutes. Pour minimiser le temps de trajet moyen dans la population, la moitié de la population doit emprunter le premier itinéraire (60 minutes), et l'autre moitié le second itinéraire (30 minutes), pour un temps de trajet moyen optimal de 45 minutes. Néanmoins, avec des agents égoïstes, personne



FIGURE 19 – Un exemple d'équilibre de Nash peu performant socialement.

n'empruntera le premier itinéraire, et le temps de trajet sera donc de 60 minutes pour tous (équilibre de Wardrop, 1952). Clairement, l'efficacité globale du trafic pourrait être améliorée en incitant les usagers à emprunter le premier itinéraire (en imposant par exemple avec une chance sur deux à un véhicule d'emprunter le premier itinéraire). Si l'espace des stratégies n'est pas ici de nature combinatoire, il le devient si l'on ne suppose plus qu'il existe seulement deux itinéraires, mais qu'on envisage tous les itinéraires possibles.

L'un des premiers objets des travaux que nous souhaitons conduire en théorie des jeux algorithmiques sera de formaliser ce type de préoccupations, puis de développer des algorithmes efficaces pour paramétrer de façon optimale les mécanismes incitatifs choisis. A ce titre, imposer à un sous-ensemble d'agents de suivre une certaine stratégie, comme dans l'exemple précédent, semble une voie intéressante à étudier. Si dans l'exemple précédent la proportion d'agents « forcés » est importante (50% en moyenne), il est d'autres cas où il suffit d'imposer une stratégie à un sous-ensemble très restreint d'agents. Considérons n agents qui veulent se connecter à un même *hub* (Anshelevich et al., 2004). Chaque agent a deux possibilités : soit il se connecte directement au hub pour un coût C (le même pour chaque agent), soit il se connecte à un *backbone* pour un très faible coût ε , et s'acquitte ensuite d'un coût additionnel C/k inversement proportionnel au nombre k d'agents utilisant le backbone, pour un coût total de $\varepsilon + C/k$. Clairement, dans cette situation, il suffit de convaincre quelques agents d'utiliser le backbone pour que tout le monde suive et que cela profite à tous.

D'autres mécanismes peuvent être envisagés pour faire converger le système vers un équilibre socialement performant, comme par exemple de disposer d'un « budget » pour modifier les données d'entrée et de l'utiliser au mieux pour obtenir le meilleur équilibre possible (pour utiliser une image dans le secteur automobile : pour un budget fixé, comment fixer les montants de la prime à la casse et du bonus écologique pour minimiser les émissions de carbone?), ou encore de supposer qu'un ou plusieurs agent(s) bienveillant(s) interviennent comme acteur du jeu dans le but d'en améliorer l'issue (c'est en fait le cas dans de nombreuses situations pratiques : les mairies préemptent les logements quand elles considèrent que le prix est trop bas, les gouvernements interviennent sur le marché dans les périodes de crise financière...).

Choix social computationnel. Dans le cadre du projet ANR ComSOC (*Computational Social Choice*), portant –comme son nom l'indique– sur le choix social computationnel (Chevalerey et al., 2007), nous souhaitons étudier des problèmes d'optimisation combinatoire où l'on ne dispose pas de valuations cardinales, mais seulement d'une échelle ordinale de valuation. Cette problématique se situe au carrefour de l'optimisation combinatoire et de la théorie du choix social car elle requiert d'être capable d'étendre une relation d'ordre sur des valuations élémentaires en une relation d'ordre sur des ensembles de valuations, ce qui a été

étudié en théorie du choix social dans le cadre du rangement d'ensembles d'opportunités (en termes de liberté de choix, de flexibilité...), ou encore d'élections de comités (Barberà et al., 2004). Lorsque l'information préférentielle dont on dispose est de nature ordinale, l'affectation arbitraire de valeurs numériques conduirait à une baisse du degré d'isomorphisme entre le problème initial et sa modélisation (Bartee, 1971). A titre d'illustration, considérons l'exemple suivant. Supposons qu'une bibliothèque universitaire souhaite s'abonner à un sous-ensemble de revues scientifiques parmi un ensemble E , les revues étant de rang A (le meilleur), B ou C (le moins bon). L'abonnement à une revue $r \in E$ a un coût $c(r)$, et la bibliothèque dispose d'un budget b . Ce problème se modélise sous la forme d'un problème de sac-à-dos où l'on cherche un sous-ensemble $S \subseteq E$ tel que $\sum_{r \in S} c(r) \leq b$, mais où la fonction objectif n'est pas explicitée. Considérons l'instance suivante de ce problème : $E = \{r_1, r_2, r_3, r_4, r_5\}$, $c(r_1) = 6$, $c(r_2) = 4$, $c(r_3) = 2$, $c(r_4) = c(r_5) = 1$, et $b = 6$. La revue r_1 est de rang A, les revues r_2, r_3 sont de rang B et les revues r_4, r_5 sont de rang C. Dans la table 7, on indique différentes valeurs numériques compatibles avec les rangs, et le choix d'abonnements qui s'ensuit. On s'aperçoit que des changements mineurs dans les valeurs numériques conduisent à des choix très différents. Ce type d'exemple justifie le soucis de développer des algorithmes spécifiquement dédiés à une information de nature ordinale.

r_1	r_2	r_3	r_4	r_5	choix
7	3	3	1	1	$\{r_1\}$
5	3	3	1	1	$\{r_2, r_3\}$
5	3	3	2	2	$\{r_3, r_4, r_5\}$

TABLE 7 – Illustration de la baisse du degré d'isomorphisme.

Pour étendre la relation d'ordre $\triangleright$ sur les valuations (les rangs A,B,C dans l'exemple, avec $A \triangleright B \triangleright C$) en une relation d'ordre $\succ$ sur les solutions (le sous-ensemble S des revues auxquelles on décide de s'abonner dans l'exemple), une règle d'extension naturelle et « minimale » en un certain sens est la suivante : $S \succ S'$ si il existe une injection $\pi : S' \rightarrow S$ telle que $\forall r \in S'$, $v(\pi(r)) \triangleright v(r)$, où $v(r)$ désigne la valuation ordinale de r . A notre connaissance, cette règle d'extension a été initialement proposée par Bossong et Schweigert (1999) (voir aussi Schweigert, 1999). Divers travaux plus récents avec des préférences ordinales utilisent également cette règle (Brams et al., 2004; Brams et King, 2005; Bouveret et al., 2010). Bossong et Schweigert ont montré que la comparaison de deux solutions selon ce principe peut se faire en temps polynomial en utilisant un algorithme de couplage dans un graphe biparti. Une autre façon de faire s'avère plus efficace et crée un pont entre optimisation ordinale et optimisation multi-objectifs. Elle consiste à remplacer chaque valuation ordinale par un vecteur comportant autant de composantes qu'il y a de valuations dans l'échelle ordinale, de la manière suivante : le rang A correspond au vecteur $(1, 1, 1)$, le rang B à $(0, 1, 1)$ et le rang C à $(0, 0, 1)$. Supposons que l'on veuille comparer les ensembles $S = \{r_2, r_3\}$ et $S' = \{r_2, r_4, r_5\}$ dans l'exemple précédent. En additionnant les vecteurs correspondants dans S et S' , on obtient $(0, 2, 2)$ pour S et $(0, 1, 3)$ pour S' . Comparer S et S' au sens de la règle d'extension décrite plus haut est équivalent à comparer ces vecteurs au sens de Pareto (ces solutions sont en l'occurrence incomparables). Rechercher les solutions non-dominées au sens de cette règle peut ainsi se faire en employant la programmation dynamique multi-objectifs, où le nombre d'objectifs est égal au nombre d'échelons ordinaux utilisés. Cette correspondance entre optimisation ordinale et multi-objectifs n'est pas spécifique à un problème particulier, et nous semble donc pouvoir être fructueusement exploitée. Nous avons d'ailleurs déjà mené un premier travail sur le problème du sac à dos ordinal (Delort et al., 2011).

Références

- M. Allais (1979) “The foundation of a positive theory of choice involving risk and a criticism of the postulate and axioms of the american school”, dans *Expected utility hypotheses and the Allais paradox*, Dordrecht, original work published in 1953, pp. 27–145.
- S. Alumur et B. Kara (2008) “Network hub location problems : The state of the art”, *European Journal of Operational Research*, tm. 190, no. 1, pp. 1–21.
- K. Andersen, K. Jörnsten et M. Lind (1996) “On bicriterion minimal spanning trees : an approximation”, *Computers and Operations Research*, tm. 23, pp. 1171–1182.
- Y. Aneja et K. Nair (1979) “Bicriteria transportation problem”, *Management Science*, tm. 25, pp. 73–78.
- E. Angel, E. Bampis et A. Kononov (2003) “On the approximate tradeoff for bicriteria batching and parallel machine scheduling problems.”, *Theor. Comput. Sci.*, tm. 306, no. 1-3, pp. 319–338.
- E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler et T. Roughgarden (2004) “The price of stability for network design with fair cost allocation”, dans *The 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 295–304.
- I. D. Aron et P. Van Hentenryck (2004) “On the complexity of the robust spanning tree problem with interval data”, *Operations Research Letters*, tm. 32, pp. 36–40.
- K. Arrow (1965) *The Theory of Risk Aversion*.
- I. Averbakh et V. Lebedev (2004) “Interval data minmax regret network optimization problems”, *Discrete Applied Mathematics*, tm. 138, pp. 289–301.
- M. Baïou et M. Balinski (2003) “Admissions and recruitment”, *The American Mathematical Monthly*, tm. 110, pp. 386–399.
- S. Barberà, W. Bossert et P. Pattanaik (2004) “Ranking sets of objects”, dans S. Barbera, P. Hammond et C. Seidl, réds., *Handbook of Utility Theory*, vol. 2, Kluwer Academic Publishers.
- E. Bartee (1971) “Problem solving with ordinal measurement”, *Management Science*, tm. 17, no. 10, pp. 622–633.
- J. Bartholdi et J. Orlin (1991) “Single transferable vote resists strategic voting”, *Social Choice and Welfare*, tm. 8, no. 4, pp. 341–354.
- J. Bartholdi III, C. Tovey et M. Trick (1989) “Voting schemes for which it can be difficult to tell who won the election”, *Social Choice and Welfare*, tm. 6, no. 2, pp. 157–165.

- H. Bast, S. Funke, D. Matijevic, P. Sanders et D. Schultes (2007) “In transit to constant time shortest-path queries in road networks”, dans *Proc. of 9th Workshop on Algorithm Engineering and Experimentation (ALENEX) 2007*.
- C. Bazgan, H. Hugot et D. Vanderpooten (2007a) “An efficient implementation for the 0-1 multi-objective knapsack problem”, dans *WEA*, pp. 406–419.
- C. Bazgan, H. Hugot et D. Vanderpooten (2007b) “A practical efficient FPTAS for the 0-1 multi-objective knapsack problem.”, dans *Arge, L., Hoffmann, M., Welzl, E. (eds.) ESA 2007 Springer LNCS 4698*, pp. 717–728.
- C. Bazgan, H. Hugot et D. Vanderpooten (2009) “Solving efficiently the 0-1 multi-objective knapsack problem”, *Computers and Operations Research*, tm. 36, no. 1, pp. 260–279.
- J.-P. Benoît (2000) “The Gibbard-Satterthwaite theorem : A simple proof”, *Economic Letters*, tm. 69, pp. 319–322.
- U. Bertelè et F. Brioschi (1972) *Nonserial dynamic programming*, Academic Press, New York, USA.
- N. Betzler, J. Guo et R. Niedermeier (2010) “Parameterized computational complexity of dodgson and young elections”, *Information and Computation*, tm. 208, pp. 165–177.
- J.-C. Billaut, D. Bouyssou et P. Vincke (2010) “Should you believe in the shangai ranking? An MCDM view”, *Scientometrics*, tm. 147, no. 1, pp. 317–341.
- J. Birge (1997) “Stochastic programming computation and applications”, *INFORMS Journal on Computing*, tm. 9, no. 2, pp. 111–133.
- G. Bitran et J. Rivera (1982) “A combined approach to solve binary multicriteria problems”, *Naval Research Logistics Quarterly*, tm. 29, pp. 181–201.
- D. Black (1948) “On the rationale of group decision-making”, *The Journal of Political Economy*, tm. 56, no. 1, pp. 23–24.
- D. Black (1958) *Theory of committees and elections*, Cambridge University Press.
- C. Blackorby et D. Donaldson (1978) “Measures of relative equality and their meaning in terms of social welfare”, *Journal of Economic Theory*, tm. 2, pp. 59–80.
- J. Blythe (1999) “Decision-theoretic planning”, *AI Magazine*, tm. 20.
- N. Boland, P. Domínguez-Marín, S. Nickel et J. Puerto (2006) “Exact procedures for solving the discrete ordered median problem”, *Computers & OR*, tm. 33, no. 11, pp. 3270–3300.
- U. Bossong et D. Schweigert (1999) “Minimal trees on ordered graphs”, Document de Travail.
- C. Boutilier (2002) “A pomdp formulation of preference elicitation problems”, dans *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI)*, pp. 239–246.
- C. Boutilier, R. Brafman, C. Domshlak, H. Hoos et D. Poole (2004) “Cp-nets : A tool for representing and reasoning with conditional ceteris paribus preference statements”, *Journal of Artificial Intelligence Research (JAIR)*, tm. 21, pp. 135–191.
- C. Boutilier, T. Dean et S. Hanks (1999) “Decision-theoretic planning : Structural assumptions and computational leverage”, *Journal of AI Research*, tm. 11, pp. 1–94.

- C. Boutilier, R. Patrascu, P. Poupart et D. Schuurmans (2006) “Constraint-based optimization and utility elicitation using the minimax decision criterion”, *Artificial Intelligence*, tm. 170, no. 8-9, pp. 686–713.
- S. Bouveret, U. Endriss et J. Lang (2010) “Fair division under ordinal preferences : Computing envy-free allocations of indivisible goods”, dans *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*, pp. 387–392.
- S. Bouveret et M. Lemaître (2007) “Fonctions d'utilité collective avec droits exogènes in-égaux”, dans *Actes des cinquièmes journées francophones Modèles Formels de l'Interaction (MFI'07)*.
- D. Bouyssou, T. Marchant et P. Perny (2009) “Social choice theory and multicriteria decision making”, dans D. Bouyssou, D. Dubois, M. Pirlot et H. Prade, réds., *Decision-making Process Concepts and Methods*, ISTE/Wiley, pp. 779–810.
- S. Brams, P. Edelman et P. Fishburn (2004) “Fair division of indivisible items”, *Theory and Decision*, tm. 5, no. 2, pp. 147–180.
- S. Brams et D. King (2005) “Efficient fair division – help the worst off or avoid envy?”, *Rationality and Society*, tm. 17, no. 4, pp. 387–421.
- P. Brucker et H. Hamacher (1989) “ k -optimal solution sets for some polynomially solvable scheduling problems”, *European Journal of Operational Research*, tm. 41, pp. 194–202.
- M. Captivo, J. Climaco, J. Figueira, E. Martins et J. Santos (2003) “Solving bicriteria 0-1 knapsack problems using a labeling algorithm”, *Comput. Oper. Res.*, tm. 30, no. 12, pp. 1865–1886.
- L. Carroll (1865) *Alice's Adventures in Wonderland*.
- U. Chajewska, D. Koller et R. Parr (2000) “Making rational decisions using adaptive utility elicitation”, dans *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI)*, pp. 363–369.
- A. Chateauneuf et J. Tallon (2002) “Diversification, convex preferences and non-empty core in the Choquet expected utility model”, *Economic Theory*, tm. 19, pp. 509–523.
- C. Chegireddy et H. Hamacher (1987) “Algorithms for finding k -best perfect matchings”, *Discrete Applied Mathematics*, tm. 18, pp. 155–165.
- Y. Chevaleyre, U. Endriss, J. Lang et N. Maudet (2007) “A short introduction to computational social choice”, dans *Proceedings of the 33rd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM-2007)*, tm. 4362 de LNCS, Springer, pp. 51–69.
- Y. Chevaleyre, U. Endriss, J. Lang et N. Maudet (2008) “Preference handling in combinatorial domains : From ai to social choice”, *AI Magazine*, tm. 29, no. 4, pp. 37–46.
- K. M. Chong (1976) “An induction theorem for rearrangements”, *Canadian Journal of Mathematics*, tm. 28, pp. 154–160.
- S. Chung, H. Hamacher, F. Maffioli et K. Murty (1993) “Note on combinatorial optimization with max-linear objective functions”, *Discrete Applied Mathematics*, tm. 42, pp. 139–145.
- V. Chvátal (1973) “On the computational complexity of finding a kernel”, *Rap. tech. CRM-300*, Centre de Recherches Mathématiques, Université de Montréal, Canada.

- M. Cohen et J.-Y. Jaffray (1980) “Rational behavior under complete ignorance”, *Econometrica*, tm. 48, no. 5, pp. 1281–1299.
- V. Conitzer, T. Sandholm et J. Lang (2007) “When are elections with few candidates hard to manipulate?”, *Journal of the ACM*, tm. 54, no. 3, p. Article 14.
- H. Corley (1985) “Efficient spanning trees”, *J. of Opt. Theory and Applications*, tm. 45, no. 3, pp. 481–485.
- P. Cramton (2006) *Combinatorial Auctions*, MIT Press.
- J. Current, M. Daskin et D. Schilling (2001) *Discrete Network Location Models*, chap. Discrete Network Location Models, Springer-Verlag, pp. 83–120.
- H. Daellenbach et C. De Kluyver (1980) “Note on multiple objective dynamic programming”, *Journal of the Operational Research Society*, tm. 31, pp. 591–594.
- G. Dantzig (1955) “Linear programming under uncertainty”, *Management Science*, tm. 1, pp. 197–206.
- G. Dantzig (1963) *Linear Programming and Extensions*, Princeton University Press, Princeton.
- C. Delort et O. Spanjaard (2010) “Using bound sets in multiobjective optimization : Application to the biobjective binary knapsack problem”, dans *9th International Symposium on Experimental Algorithms (SEA 2010)*, tm. 6049 de *Lecture Notes in Computer Science*, pp. 253–265.
- C. Delort et O. Spanjaard (2011) “Yet another two-phase method for the biobjective assignment problem”, dans *21st International Conference on Multiple Criteria Decision Making (MCDM 2011)*, Jyvaskyla, Finland.
- C. Delort, O. Spanjaard et P. Weng (2011) “Committee selection with a weight constraint based on a pairwise dominance relation”, dans *2nd International Conference on Algorithmic Decision Theory, ADT-11*, tm. 6992, *Lecture Notes in Artificial Intelligence*, Springer-Verlag, pp. 28–41.
- C. Dodgson (1876) *A method for taking votes on more than two issues*, Clarendon Press. Reproduit dans (Black, 1958) avec une discussion.
- C. Domshlak, S. Genaim et R. Brafman (2001) “Preference-based configuration of web page content”, dans *Proceedings of the 17th international joint conference on Artificial intelligence (IJCAI)*, pp. 1451–1456.
- R. G. Downey et M. R. Fellows (1999) *Parameterized Complexity*, Springer.
- J. Edmonds (1970) *Submodular functions, matroids and certain polyhedra*, International conference on combinatorics, pp. 69–87.
- M. Ehrgott (2005) *Multicriteria Optimization, second edition*, Springer.
- M. Ehrgott et X. Gandibleux (2000) “A survey and annotated bibliography of multiobjective combinatorial optimization”, *OR Spektrum*, tm. 22, pp. 425–460.
- M. Ehrgott et X. Gandibleux (2004) “Approximative solution methods for multiobjective combinatorial optimization”, *Journal of the Spanish Statistical and Operations Research Society*, tm. 12, no. 1, pp. 1–88.

- M. Ehrgott et X. Gandibleux (2007) “Bound sets for biobjective combinatorial optimization problems”, *Computers & Operations Research*, tm. 34, no. 9, pp. 2674–2694.
- M. Ehrgott et A. Skriver (2003) “Solving biobjective combinatorial max-ordering problems by ranking methods and a two-phases approach”, *European Journal of Operational Research*, tm. 147, pp. 657–664.
- J. El Haddad et O. Spanjaard (2009) “Composition de services web et équité vis-à-vis des utilisateurs finaux”, dans *10ème Congrès de la ROADEF*, pp. 251–252.
- D. Ellsberg (1961) “Risk, ambiguity and the savage axioms”, *Quarterly Journal of Economics*, tm. 75, pp. 643–669.
- V. Emelichev et V. Perepelitsa (1988) “Multiobjective problems on the spanning trees of a graph”, *Soviet Math. Dokl.*, tm. 37, no. 1, pp. 114–117.
- D. Eppstein (1998) “Finding the k shortest paths”, *SIAM Journal on Computing*, tm. 28, no. 2, pp. 652–673.
- T. Erlebach, H. Kellerer et U. Pferschy (2002) “Approximating multiobjective knapsack problems”, *Manag. Science*, tm. 48, no. 12, pp. 1603–1612.
- B. Escoffier, L. Gourvès, J. Monnot et O. Spanjaard (2010) “Two-stage stochastic matching and spanning tree problems : polynomial instances and approximation”, *European Journal of Operational Research*, tm. 205, no. 1, pp. 19–30.
- B. Escoffier, J. Lang et M. Öztürk (2008a) “Single-peaked consistency and its complexity”, dans *Proc. of the European Conference on Artificial Intelligence (ECAI 2008)*, IOS Press, pp. 366–370.
- B. Escoffier, J. Monnot et O. Spanjaard (2008b) “Some tractable instances of interval data minmax regret problems”, *Operations Research Letters*, tm. 36, pp. 424–429.
- B. Escoffier, J. Monnot et O. Spanjaard (2008c) “Some tractable instances of interval data minmax regret problems : bounded distance from triviality (short version)”, dans *34th International Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM-08*, LNCS 4910, Springer-Verlag, pp. 280–291.
- P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra et J. Rothe (2009) “A richer understanding of the complexity of election systems”, dans S. Ravi et S. Shukla, réds., *Fundamental Problems in Computing : Essays in Honor of Professor Daniel J. Rosenkrantz*, Springer, pp. 375–406.
- H. Fargier, G. Jeantet et O. Spanjaard (2011) “Resolute choice in sequential decision problems with multiple priors”, dans *22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, Barcelona, Catalonia (Spain), pp. 2120–2125.
- P. Fishburn (1970) *Utility theory for decision making*, Wiley.
- H. Fouchal, X. Gandibleux et F. Lehuédé (2011) “Preferred solutions computed with a label setting algorithm based on choquet integral for multi-objective shortest paths”, dans *IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making*, pp. 143–150.
- S. Fujishige (2005) *Submodular functions and Optimization*, tm. Annals of discrete mathematics, 58, Elsevier Science Publishers.

- H. Gabow (1977) “Two algorithms for generating weighted spanning trees in order”, *SIAM J. Comput.*, tm. 6, no. 1, pp. 139–150.
- T. Gajdos (2001) “Les fondements axiomatique de la mesure normative des inégalités”, *Revue d’Economie Politique*, tm. 5, pp. 683–720.
- L. Galand, J. Lesca et P. Perny (2011) “Multiobjective dynamic programming versus linear programming for compromise search with choquet integral”, dans *International Conference on Multiple Criteria Decision Making*, Jyväskylä, Finland, June 13-17, 2011, p. 175.
- L. Galand et P. Perny (2006) “Search for compromise solutions in multiobjective state space graphs”, dans *Proc. of the 17th European Conference on Artificial Intelligence*, pp. 93–97.
- L. Galand, P. Perny et O. Spanjaard (2009) “A branch and bound algorithm for Choquet optimization in multicriteria problems”, dans M. Ehrgott, B. Naujoks, T. Stewart et J. Wallenius, réds., *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems, MCDM-08*, tm. 634 de *Lecture Notes in Economics and Mathematical Systems*, 10 pages.
- L. Galand, P. Perny et O. Spanjaard (2010) “Choquet-based optimisation in multiobjective shortest path and spanning tree problems”, *European Journal of Operational Research*, tm. 204, no. 2, pp. 303–315.
- L. Galand et O. Spanjaard (2007a) “Deux approches complémentaires pour un problème d’arbre couvrant robuste”, dans *8ème Congrès de la ROADEF*, Presses Universitaires de Grenoble, pp. 129–137.
- L. Galand et O. Spanjaard (2007b) “OWA-based search in state space graphs with multiple cost functions”, dans *20th International Florida Artificial Intelligence Research Society Conference, FLAIRS-07*, AAAI Press, pp. 86–91.
- L. Galand et O. Spanjaard (2012) “Exact algorithms for OWA-optimization in multiobjective spanning tree problems”, *Computers & Operations Research*, tm. 39, pp. 1540–1554.
- D. Gale et L. Shapley (1962) “College admissions and the stability of marriage”, *The American Mathematical Monthly*, tm. 69, no. 1, pp. 9–15.
- M. Garey et D. Johnson (1979) *Computers and Intractability*, Freeman and Co., New York.
- N. Garg, T. Kavitha, A. Kumar, K. Mehlhorn et J. Mestre (2010) “Assigning papers to referees”, *Algorithmica*, tm. 58, pp. 119–136.
- J.-P. Gayant (2001) *Risque et décision*, Vuibert.
- A. Gibbard (1973) “Manipulation of voting schemes : A general result”, *Econometrica*, tm. 41, no. 4, pp. 587–601.
- I. Gilboa et M. Marinacci (2011) “Ambiguity and the bayesian paradigm”, *Working Papers 379*, IGIER (Innocenzo Gasparini Institute for Economic Research), Bocconi University.
- I. Gilboa et D. Schmeidler (1989) “Maxmin expected utility with non-unique prior”, *Journal of Economic Theory*, tm. 18, pp. 141–153.
- A. Goldberg (2007) “Point-to-point shortest path algorithms with preprocessing”, dans *33rd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, pp. 88–102.

- A. Goldberg et C. Harrelson (2005) “Computing the shortest path : search meets graph theory”, dans *SODA*, pp. 156–165.
- B. Golden et P. Perny (2010) “Infinite order Lorenz dominance for fair multiagent optimization”, dans *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 383–390.
- C. Gonzales et P. Perny (2004) “GAI networks of utility elicitation”, dans *Proceedings of the 9th International Conference on the Principles of Knowledge Representation and Reasoning*, pp. 224–234.
- M. Grabisch (1996) “The application of fuzzy integrals in multicriteria decision making”, *European Journal of Operational Research*, tm. 89, pp. 445–456.
- M. Grabisch (1997) “ k -order additive discrete fuzzy measures and their representation”, *Fuzzy Sets and Systems*, tm. 92, pp. 167–189.
- J. Granat et F. Guerriero (2003) “The interactive analysis of the multicriteria shortest path problem by the reference point method”, *European Journal of Operational Research*, tm. 151, pp. 103–118.
- J. Guo, F. Hüffner et R. Niedermeier (2004) “A structural view on parameterizing problems : Distance from triviality”, dans *IWPEC 2004*, tm. 3162 de *LNCS*, pp. 162–173.
- G. Haeringer et V. Iehlé (2010) “Enjeux stratégiques du concours de recrutement des enseignants-chercheurs”, *Revue économique*, tm. 61, pp. 697–721.
- H. Hamacher (1995) “A note on k -best network flows”, *Annals of Operations Research*, tm. 57, pp. 65–72.
- H. Hamacher et G. Ruhe (1994) “On spanning tree problems with multiple objectives”, *Annals of Operations Research*, tm. 52, pp. 209–230.
- J. Handa (1977) “Risk, probabilities and a new theory of cardinal utility”, *Journal of Political Economics*, tm. 85, pp. 97–122.
- P. Hansen (1980) “Bicriterion path problems”, dans G. Fandel et T. Gal, réds., *Multicriteria Decision Making*.
- G. H. Hardy, J. E. Littlewood et G. Pólya (1934) *Inequalities*, Cambridge University Press.
- R. Howard et J. Matheson (1984) “Influence diagrams”, dans *Readings on the Principles and Applications of Decision Analysis*, pp. 721–762.
- N. Immorlica, D. R. Karger, M. Minkoff et V. S. Mirrokni (2004) “On the costs and benefits of procrastination : approximation algorithms for stochastic combinatorial optimization problems”, dans *Proc. of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2004)*, p. 691–700.
- J. Jaffray et M. Jeleva (2007) “Information processing under imprecise risk with the Hurwicz criterion”, dans *5th International Symposium on Imprecise Probability : Theories and Applications*, pp. 233–242.
- J.-Y. Jaffray et T. Nielsen (2006) “An operational approach to rational decision making based on rank dependent utility”, *European J. of Operational Research*, tm. 169, no. 1, pp. 226–246.

- G. Jeantet et O. Spanjaard (2008) “Rank-dependent probability weighting in sequential decision problems under uncertainty”, dans *International Conference on Automated Planning and Scheduling, ICAPS-08*, AAAI Press, pp. 148–155.
- G. Jeantet et O. Spanjaard (2009) “Optimisation de l’utilité espérée dépendant du rang dans les diagrammes d’influence”, dans *10ème Congrès de la Société Française de Recherche Opérationnelle et d’Aide à la Décision*.
- G. Jeantet et O. Spanjaard (2010) “Une approche de choix résolu au sens de jaffray dans les arbres de décision munis de probabilités imprécises”, dans *Onzième Congrès de la Société Française de Recherche Opérationnelle et d’Aide à la Décision (ROADEF 2010)*, ENAC - Toulouse, pp. 141–154.
- G. Jeantet et O. Spanjaard (2011) “Computing rank dependent utility in graphical models”, *Artificial Intelligence Journal*, tm. 175, pp. 1366–1389.
- F. Jensen, F. Jensen et S. Dittmer (1994) “From influence diagrams to junction trees”, dans *Proc. of the Tenth Annual Conf. on Uncertainty in Artificial Intelligence (UAI)*, pp. 367–373.
- D. Kahneman et A. Tversky (1979) “Prospect theory : an analysis of decision under risk”, *Econometrica*, tm. 47, pp. 263–291.
- F. Kappel et A. Kuntsevich (2000) “An implementation of shor’s r -algorithm”, *Computational Optimization and Applications*, tm. 15, pp. 193–205.
- D. Kikuti, F. G. Cozman et R. S. Filho (2011) “Sequential decision making with partially ordered preferences”, *Artificial Intelligence*, tm. 175, pp. 1346–1365.
- G. Kiziltan et E. Yucaoglu (1983) “An algorithm for multiobjective zero-one linear programming”, *Management Science*, tm. 29, no. 12, pp. 1444–1453.
- K. Klamroth et M. Wiecek (2000) “Dynamic programming approaches to the multiple criteria knapsack problem”, *Naval Research Logistics*, tm. 47, pp. 57–76.
- J. Knowles et D. Corne (2001) “A comparison of encodings and algorithms for multiobjective spanning tree problems”, dans *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pp. 544–551.
- P. Kouvelis et G. Yu (1997) *Robust discrete optimization and its applications*, Kluwer Academic Publisher.
- M. Laumanns, L. Thiele, K. Deb et E. Zitzler (2002) “Combining convergence and diversity in evolutionary multiobjective optimization”, *Evolutionary Computation*, tm. 10, no. 3, pp. 263–282.
- J. Lesca et P. Perny (2010) “LP solvable models for multiagent fair allocation problems”, dans *Proc. of the European Conference on Artificial Intelligence (ECAI 2010)*, IOS Press, pp. 387–392.
- I. Levi (1974) “On indeterminate probabilities”, *Journal of Philosophy*, tm. 71, no. 13, pp. 391–418.
- R. Luce et H. Raiffa (1957) *Games and Decisions*, Wiley, New York.

- T. Magnanti et L. Wolsey (1995) *Optimal trees*, tm. 7, North-Holland, pp. 503–615.
- L. Mandow et J. P. de la Cruz (2003) “Multicriteria heuristic search”, *European Journal of Operational Research*, tm. 150, pp. 253–280.
- O. Marcotte et R. Soland (1986) “An interactive branch-and-bound algorithm for multiple criteria optimization”, *Management Science*, tm. 32, no. 1, pp. 61–75.
- W. Marshall et I. Olkin (1979) *Inequalities : Theory of Majorization and its Applications*, Academic Press, London.
- P. Martin et D. Shmoys (1996) “A new approach to computing optimal schedules for the job shop scheduling problem”, dans *Proceedings of the Fifth international IPCO conference*, tm. LNCS 1084, pp. 389–403.
- G. Mavrotas et D. Diakoulaki (1998) “A branch and bound algorithm for mixed zero-one multiple objective linear programming”, *European Journal of Operational Research*, tm. 107, pp. 530–541.
- E. McClennen (1990) *Rationality and Dynamic choice : Foundational Explorations*, Cambridge University Press.
- H. Moulin (1988) *Axioms of cooperative decision making*, Monograph of the Econometric Society, Cambridge University Press.
- V. Mousseau (2003) *Elicitation des préférences pour l’aide multicritère à la décision*, Thèse de doctorat, Mémoire présenté en vue de l’obtention de l’habilitation à diriger des recherches, Université Paris-Dauphine.
- I. Murthy et S. Her (1992) “Solving min-max shortest-path problems on a network”, *Naval Research Logistics*, tm. 39, pp. 669–683.
- N. Nisan, T. Roughgarden, E. Tardos et V. Vazirani (2007) *Algorithmic Game Theory*, Cambridge University Press, New York, USA.
- W. Ogryczak (2009) “Inequality measures and equitable locations”, *Annals of Operations Research*, tm. 167, pp. 61–86.
- W. Ogryczak et T. Sliwinski (2003) “On solving linear programs with the ordered weighted averaging objective”, *European Journal of Operational Research*, tm. 127, no. 1, pp. 80–91.
- C. Papadimitriou et M. Yannakakis (2000) “The complexity of tradeoffs, and optimal access of web sources”, dans *41st Annual IEEE Symposium on Foundations of Computer Science FOCS*, pp. 86–92.
- J. Pearl (1988) *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*, Morgan Kaufman Publishers.
- F. Perea et J. Puerto (2007) “Dynamic programming analysis of the TV game “Who wants to be a millionaire?””, *European Journal of Operational Research*, tm. 183, pp. 805–811.
- P. Perny (2000) *Modélisation des préférences, agrégation multicritère et systèmes d’aide à la décision*, Habilitation à diriger des recherches, Université Paris 6.

- P. Perny et O. Spanjaard (2003) “An axiomatic approach to robustness in search problems with multiple scenarios”, dans *Proceedings of the 19th conference on Uncertainty in Artificial Intelligence*, Acapulco, Mexico, pp. 469–476.
- P. Perny, O. Spanjaard et L.-X. Storme (2006) “A decision-theoretic approach to robust optimization in multivalued graphs”, *Annals of Operations Research*, tm. 147, no. 1, pp. 317–341.
- P. Perny, O. Spanjaard et L.-X. Storme (2007) “State space search for risk-averse agents”, dans *20th International Joint Conference on Artificial Intelligence, IJCAI-07*, pp. 2353–2358.
- A. Pigou (1920) *The Economics of Welfare*, Macmillan.
- J. Pratt (1964) “Risk aversion in the small and in the large”, *Econometrica*, tm. 32, no. 1, pp. 122–136.
- A. Punnen et Y. Aneja (1995) “Minimax combinatorial optimization”, *European Journal of Operational Research*, tm. 81, pp. 634–643.
- M. Puterman (1994) *Markov Decision Processes : Discrete Stochastic Dynamic Programming*, John Wiley & Sons.
- J. Quiggin (1982) “A theory of anticipated utility”, *Journal of Economic Behavior and Organisation*, tm. 3, no. 4, pp. 323–343.
- J. Quiggin (1993) *Generalized expected utility theory : the rank-dependent model*, Kluwer Academic Publishers, Boston.
- H. Raiffa (1968) *Decision Analysis : Introductory Lectures on Choices under Uncertainty*, Addison-Wesley.
- R. Ramos, S. Alonso, J. Sicilia et C. Gonzalez (1998) “The problem of the optimal biobjective spanning tree”, *European Journal of Operational Research*, tm. 111, pp. 617–628.
- C. ReVelle et H. Eiselt (2005) “Location analysis : A synthesis and survey”, *European Journal of Operational Research*, tm. 165, pp. 1–19.
- E. Rosinger (1991) “Beyond preference information based multiple criteria decision making”, *European Journal of Operational Research*, tm. 53, no. 2, pp. 217–227.
- B. Roy et D. Bouyssou (1993) *Aide multicritère à la décision : méthodes et cas*, Economica, Paris.
- S. Ruzika et H. Hamacher (2009) “A survey on multiple objective minimum spanning tree problems”, tm. 5515, Springer, pp. 104–116.
- N. Sahinidis (2004) “Optimization under uncertainty : state-of-the-art and opportunities”, *Computers and Chemical Engineering*, tm. 28, pp. 971–983.
- T. Sandholm et C. Boutilier (2006) “Preference elicitation in combinatorial auctions”, dans Cramton, Shoham et Steinberg, réds., *Combinatorial Auctions*, chap. 10, MIT Press.
- M. Satterthwaite (1975) “Strategy-proofness and arrow’s conditions”, *Journal of Economic Theory*, tm. 10, pp. 187–217.
- L. Savage (1954) *The Foundations of Statistics*, John Wiley and Sons, New York, ,second revised edition, 1972, Dover, NY.

- D. Schmeidler (1986) “Integral representation without additivity”, *Proceedings of the American Mathematical Society*, tm. 97, no. 2, pp. 255–261.
- D. Schmeidler (1989) “Subjective probability and expected utility without additivity”, *Econometrica*, tm. 57, pp. 571–587.
- D. Schweigert (1999) “Ordered graphs and minimal spanning trees”, *Foundations of Computing and Decision Sciences*, tm. 24, no. 4, pp. 219–229.
- T. Seidenfeld (1988) “Decision theory without independence or without ordering”, *Econ. and Phil.*, tm. 4, no. 2, pp. 267–290.
- T. Seidenfeld (2004) “A contrast between two decision rules for use with (convex) sets of probabilities : G-maximin versus e-admissibility”, *Synthèse*, tm. 140, pp. 69–88.
- P. Serafini (1986) “Some considerations about computational complexity for multiobjective combinatorial problems”, dans J. Jahn et W. Krabs, réds., *Recent advances and historical development of vector optimization*, tm. 294 de *Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, Berlin.
- R. Shachter (1986) “Evaluating influence diagrams”, *Operations Research*, tm. 34, pp. 871–882.
- L. Shapley (1971) “Cores of convex games”, *International Journal of Game Theory*, tm. 1, pp. 11–22.
- A. Shorrocks (1983) “Ranking income distributions”, *Economica*, tm. 50, pp. 3–17.
- F. Sourd et O. Spanjaard (2008) “A multi-objective branch-and-bound framework. application to the bi-objective spanning tree problem”, *INFORMS Journal on Computing*, tm. 20, no. 3, pp. 472–484.
- F. Sourd, O. Spanjaard et P. Perny (2006) “Multi-objective branch-and-bound. application to the bi-objective spanning tree problem”, dans *7th International Conference on Multi-Objective Programming and Goal Programming (MOPGP)*, p. 5 pages.
- O. Spanjaard (2003) *Exploitation de préférences non-classiques dans les problèmes combinatoires : modèles et algorithmes pour les graphes*, Thèse de doctorat, Université Paris-Dauphine.
- S. Steiner et T. Radzik (2008) “Computing all efficient solutions of the biobjective minimum spanning tree problem”, *Computers & Operations Research*, tm. 35, pp. 198–211.
- R. Steuer et E. Choo (1983) “An interactive weighted tchebycheff procedure for multiple objective programming”, *Mathematical Programming*, tm. 26, p. 326–344.
- B. Stewart et C. White III (1991) “Multiobjective A*”, *Journal of the Association for Computing Machinery*, tm. 38, no. 4, pp. 775–814.
- S. Subbarayan, R. Jensen, T. Hadzic, H. Andersen, J. Møller et H. Hulgaard (2004) “Comparing two implementations of a complete and backtrack-free interactive configurator”, dans *Proceedings of the CP-04 Workshop on CSP Techniques with Immediate Application*, pp. 97–111.
- M. Trick (1989) “Recognizing single-peaked preferences on a tree”, *Mathematical Social Sciences*, tm. 17, no. 3, pp. 329–334.

- G. Tsaggouris et C. Zaroliagis (2009) “Multiobjective optimization : Improved FPTAS for shortest paths and non-linear objectives with applications”, *Theory of Computing Systems*, tm. 45, no. 1, pp. 162–186.
- J. Uckelman, Y. Chevaleyre, U. Endriss et J. Lang (2009) “Representing utility functions via weighted goals”, *Mathematical Logic Quarterly*, tm. 55, no. 4, pp. 341–361.
- B. Villareal et M. Karwan (1981) “Multicriteria integer programming : A (hybrid) dynamic programming recursive approach”, *Mathematical Programming*, tm. 21, pp. 204–223.
- M. Visée, J. Teghem, M. Pirlot et B. Ulungu (1998) “Two-phases method and branch and bound procedures to solve the bi-objective knapsack problem”, *J. of Global Optimization*, tm. 12, no. 2, pp. 139–155.
- J. von Neuman et O. Morgenstern (1947) *Theory of games and economic behaviour*, Princeton University Press, Princeton, 2nd edition.
- P. Walley (1991) *Statistical reasoning with imprecise probabilities*, tm. 91 de *Monographs on statistics and applied probability*, Chapman and Hall.
- A. Warburton (1987) “Approximation of pareto optima in multiple-objective, shortest-path problems”, *Operations research*, tm. 35, no. 1, pp. 70–79.
- J. Wardrop (1952) “Some theoretical aspects of road traffic research”, dans *Proceedings of the Institution of Civil Engineers, Part II*, tm. 1, pp. 325–378.
- R. Yager (1988) “On ordered weighted averaging aggregation operators in multicriteria decision making”, dans *IEEE Trans. Systems, Man and Cybern.*, tm. 18, pp. 183–190.
- G. Yu (1998) “Min-max optimization of several classical discrete optimization problems”, *Journal of Optimization Theory and Applications*, tm. 98, no. 1, pp. 221–242.
- G. Zhou et M. Gen (1999) “Genetic algorithm approach on multi-criteria minimum spanning tree problem”, *European Journal of Operational Research*, tm. 114, pp. 141–152.

Publications

Articles publiés dans des revues internationales avec comité de lecture

- L. Galand et O. Spanjaard (2012) “Exact algorithms for OWA-optimization in multiobjective spanning tree problems”, *Computers & Operations Research*, vol. 39, pp. 1540–1554.
- G. Jeantet et O. Spanjaard (2011) “Computing rank dependent utility in graphical models”, *Artificial Intelligence Journal*, vol. 175, pp. 1366–1389.
- B. Escoffier, L. Gourvès, J. Monnot et O. Spanjaard (2010) “Two-stage stochastic matching and spanning tree problems : polynomial instances and approximation”, *European Journal of Operational Research*, vol. 205, n. 1, pp. 19–30.
- L. Galand, P. Perny et O. Spanjaard (2010) “Choquet-based optimisation in multiobjective shortest path and spanning tree problems”, *European Journal of Operational Research*, vol. 204, n. 2, pp. 303–315.
- B. Escoffier, J. Monnot et O. Spanjaard (2008) “Some tractable instances of interval data minmax regret problems”, *Operations Research Letters*, vol. 36, pp. 424–429.
- F. Sourd et O. Spanjaard (2008) “A multi-objective branch-and-bound framework. application to the bi-objective spanning tree problem”, *INFORMS Journal on Computing*, vol. 20, n. 3, pp. 472–484.
- P. Perny, O. Spanjaard et L.-X. Storme (2006) “A decision-theoretic approach to robust optimization in multivalued graphs”, *Annals of Operations Research*, vol. 147, n. 1, pp. 317–341.
- P. Perny et O. Spanjaard (2005) “A preference-based approach to spanning trees and shortest paths problems”, *European Journal of Operational Research*, vol. 162, pp. 584–601.
- J. Monnot et O. Spanjaard (2003) “Bottleneck shortest paths on a partially ordered scale”, *4OR - A Quarterly Journal of Operations Research*, vol. 1, n. 3, pp. 225–241.

Articles publiés dans des conférences internationales avec comité de lecture

- C. Delort, O. Spanjaard et P. Weng (2011) “Committee Selection with a Weight Constraint Based on a Pairwise Dominance Relation”, dans *2nd International Conference on Algorithmic Decision Theory, ADT-11*, vol. 6992 de *Lecture Notes in Artificial Intelligence*, Springer-Verlag, pp. 28–41.
- B. Escoffier, L. Gourvès, K.T. Nguyen, F. Pascual et O. Spanjaard (2011) “Strategy-proof Mechanisms for Facility Location Games with Many Facilities”, dans *2nd International Conference on Algorithmic Decision Theory, ADT-11*, vol. 6992 de *Lecture Notes in Artificial Intelligence*, Springer-Verlag, pp. 67–81.
- H. Fargier, G. Jeantet et O. Spanjaard (2011) “Resolute choice in sequential decision problems with multiple priors”, dans *22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, Barcelona, Catalonia (Spain), pp. 2120–2125.

- C. Delort et O. Spanjaard (2010) “Using bound sets in multiobjective optimization : Application to the biobjective binary knapsack problem”, dans *9th International Symposium on Experimental Algorithms (SEA 2010)*, vol. 6049 de *Lecture Notes in Computer Science*, pp. 253–265.
- L. Galand, P. Perny et O. Spanjaard (2009) “A branch and bound algorithm for Choquet optimization in multicriteria problems”, dans M. Ehrgott, B. Naujoks, T. Stewart et J. Wallenius, eds., *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems, MCDM-08*, vol. 634 de *Lecture Notes in Economics and Mathematical Systems*, 10 pages.
- G. Jeantet et O. Spanjaard (2009) “Optimizing the Hurwicz criterion in decision trees with imprecise probabilities”, dans F. Rossi et A. Tsoukias, eds., *1st International Conference on Algorithmic Decision Theory, ADT-09*, Lecture Notes in Artificial Intelligence, Springer-Verlag, pp. 340–352.
- B. Escoffier, J. Monnot et O. Spanjaard (2008) “Some tractable instances of interval data minmax regret problems : bounded distance from triviality (short version)”, dans *34th International Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM-08*, LNCS 4910, Springer-Verlag, pp. 280–291.
- G. Jeantet et O. Spanjaard (2008) “Rank-dependent probability weighting in sequential decision problems under uncertainty”, dans *International Conference on Automated Planning and Scheduling, ICAPS-08*, AAAI Press, pp. 148–155.
- P. Perny et O. Spanjaard (2008) “Near admissible algorithms for multiobjective search”, dans *18th European Conference on Artificial Intelligence ECAI-08*, IOS Press, pp. 490–494.
- L. Galand et O. Spanjaard (2007) “OWA-based search in state space graphs with multiple cost functions”, dans *20th International Florida Artificial Intelligence Research Society Conference, FLAIRS-07*, AAAI Press, pp. 86–91.
- P. Perny, O. Spanjaard et L.-X. Storme (2007) “State space search for risk-averse agents”, dans *20th International Joint Conference on Artificial Intelligence, IJCAI-07*, pp. 2353–2358.
- P. Perny, O. Spanjaard et P. Weng (2005) “Algebraic markov decision processes”, dans *19th International Joint Conference on Artificial Intelligence, IJCAI-05*, pp. 1372–1377.
- P. Perny et O. Spanjaard (2003) “An axiomatic approach to robustness in search problems with multiple scenarios”, dans *Proceedings of the 19th conference on Uncertainty in Artificial Intelligence, UAI-03*, Acapulco, Mexico, pp. 469–476.
- P. Perny and O. Spanjaard (2002) “On preference-based search in state space graphs”, dans *Proceedings of the 18th National Conference on Artificial Intelligence, AAAI-02*, pp. 751–756.

Articles publiés dans des conférences nationales avec comité de lecture

- G. Jeantet et O. Spanjaard (2010) “Une approche de choix résolu au sens de Jaffray dans les arbres de décision munis de probabilités imprécises”, dans *Onzième Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2010)*, ENAC - Toulouse, pp. 141–154.

- G. Jeantet et O. Spanjaard (2009) “Choix résolu et utilité espérée dépendant du rang dans les diagrammes d’influence”, dans *Cinquièmes Journées Francophones Modèles formels de l’interaction (MFI-09)*, pp. 181–192, isbn : 978-2-95-276302.
- G. Jeantet et O. Spanjaard (2008) “Approche algorithmique de la recherche d’une stratégie rdu-optimale dans un arbre de décision”, dans *9ème Congrès de la ROADEF*, Presses Universitaires de l’Université Blaise Pascal, pp. 79–94. *Best long paper award ROADEF-08*.
- L. Galand et O. Spanjaard (2007) “Deux approches complémentaires pour un problème d’arbre couvrant robuste”, dans *8ème Congrès de la ROADEF*, Presses Universitaires de Grenoble, pp. 129–137.

Chapitres de livre

- B. Escoffier et O. Spanjaard (2010) “Dynamic programming”, dans V. T. Paschos, ed., *Concepts of Combinatorial Optimization*, chap. 4, ISTE – Wiley, pp. 71–98.
- B. Escoffier et O. Spanjaard (2005) “Programmation dynamique”, dans V. T. Paschos, ed., *Optimisation combinatoire (Volume 1 : concepts fondamentaux)*, chap. 4, Hermès, pp. 95–124.

Communications dans des conférences internationales sur résumé

- C. Delort et O. Spanjaard (2011) “Yet another two-phase method for the biobjective assignment problem”, dans *21st International Conference on Multiple Criteria Decision Making (MCDM 2011)*, Jyväskylä, Finland.
- L. Galand, P. Perny et O. Spanjaard (2008) “Optimization of the choquet integral in multi-criteria combinatorial problems”, dans *19th International Conference on Multiple Criteria Decision Making (MCDM 2008)*, Auckland, New Zealand.
- F. Sourd, O. Spanjaard et P. Perny (2006) “Multi-objective branch and bound. application to the bi-objective spanning tree problem”, dans *7th International Conference in Multi-Objective Programming and Goal Programming*, Tours, France.
- P. Perny et O. Spanjaard (2001) “Sur la recherche des solutions préférées dans les problèmes combinatoires multicritères”, dans *3èmes journées francophones de recherche opérationnelle FRANCORO III*, Québec, Canada.

Communications dans des conférences nationales sur résumé

- C. Delort, O. Spanjaard et P. Weng (2011) “Sélection d’un comité fondée sur une classification ordinale des individus”, dans *12ème Congrès de la Société Française de Recherche Opérationnelle et d’Aide à la Décision (ROADEF 2011)*, Saint-Etienne.
- B. Escoffier, L. Gourvès, K. Nguyen, F. Pascual et O. Spanjaard (2011) “Algorithmes à véracité garantie pour le placement d’installations sur une ligne”, dans *12ème Congrès de la Société*

Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2011), Saint-Etienne.

- H. Fargier, G. Jeantet et O. Spanjaard (2011) “Optimisation de l'utilité espérée pessimiste dans les arbres de décision avec a priori multiples”, dans *12ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2011)*, Saint-Etienne.
- J. E. Haddad et O. Spanjaard (2009) “Composition de services web et équité vis-à-vis des utilisateurs finaux”, dans *10ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2009)*, Nancy.
- G. Jeantet et O. Spanjaard (2009) “Optimisation de l'utilité espérée dépendant du rang dans les diagrammes d'influence”, dans *10ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2009)*, Nancy.
- O. Spanjaard (2005) “Agrégation ordinale dans les problèmes combinatoires”, dans *6ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2005)*, Tours.
- O. Spanjaard (2003) “énumération des arbres couvrants efficaces d'un graphe à valuations multicritères”, dans *5ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2003)*, Avignon.
- J. Monnot et O. Spanjaard (2002) “Plus court chemin bottleneck sur une échelle qualitative partiellement ordonnée”, dans *4ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2002)*, Paris.

Autres

- O. Spanjaard (2005) “Décision et optimisation combinatoire”, dans *Bulletin de la Société Française de Recherche Opérationnelle et d'Aide à la Décision*, pp. 3–6.
- O. Spanjaard (2003) *Exploitation de préférences non-classiques dans les problèmes combinatoires : modèles et algorithmes pour les graphes*, Thèse de doctorat, LAMSADE - Université Paris Dauphine.
- P. Perny et O. Spanjaard (2002) “Modélisation des préférences et recherche dans les graphes d'états”, dans Cépaduès-éditions, ed., *Actes des 2e assises nationales du GdR I3*, pp. 161–171.

Résumé

Les travaux présentés dans ce mémoire portent sur l'étude de problèmes de décision sous l'angle de l'algorithmique. La plupart des travaux réalisés en théorie de la décision jusqu'à récemment focalisant sur la caractérisation de solutions préférées, l'aspect computationnel (i.e., la calculabilité effective de ces solutions) est souvent relégué au second plan. C'est précisément ce deuxième aspect qui est au coeur des thèmes de recherche abordés ici. Ceux-ci se situent par conséquent au carrefour de plusieurs domaines : algorithmique, décision multicritère, décision dans l'incertain, optimisation combinatoire, intelligence artificielle. Le mémoire est organisé en quatre parties. La première partie est une introduction aux problématiques abordées en *théorie de la décision algorithmique*. Divers modèles décisionnels pour la décision en présence de points de vue multiples et/ou dans l'incertain sont présentés dans la deuxième partie. Les deux parties suivantes portent respectivement sur la résolution de problèmes combinatoires multi-objectifs et sur l'optimisation combinatoire dans l'incertain, cette dernière dénomination couvrant l'optimisation combinatoire robuste ainsi que l'optimisation de critères de décision non-conséquentialistes dans des problèmes de décision séquentielle dans l'incertain. Le mémoire s'achève par une présentation des directions de recherche futures.

Mots-clés : Aide à la décision, recherche opérationnelle, intelligence artificielle, théorie de la décision, optimisation combinatoire, théorie de la décision algorithmique, optimisation combinatoire multi-objectifs, optimisation discrète robuste, décision séquentielle dans l'incertain.

Abstract

The works presented here concern the study of decision problems in terms of algorithms. Most works in decision theory until recently focusing on the characterization of preferred solutions, the computational aspect (i.e., the effective computability of solutions) is often relegated to the background. It is precisely this second aspect that is central to the research topics discussed here. They are therefore at the crossroads of several areas : algorithms, multicriteria decision making, decision under uncertainty, combinatorial optimization, artificial intelligence. The presentation is organized into four sections. The first section is an introduction to the issues discussed in *algorithmic decision theory*. Various decision models for multiple criteria decision making and decision making under uncertainty are presented in the second section. The next two sections deal respectively with multi-objective combinatorial optimization and optimization under uncertainty, this latter denomination covering robust discrete optimization as well as the optimization of non-consequentialist decision criteria in sequential decision problems under uncertainty. The brief is concluded by a presentation of future research directions.

Keywords : Decision aiding, operations research, artificial intelligence, decision theory, combinatorial optimization, algorithmic decision theory, multi-objective combinatorial optimization, robust discrete optimization, sequential decision making under uncertainty.