

Sommaire

Remerciements	4
Résumé	6
Abstract	7
Sommaire	8
Introduction	11
Chapitre 1 : Aide à la décision Multicritère / Multiattribut.....	15
1.1. Historique de l'aide à la décision	15
1.2. Aide à la décision	20
1.2.1. Aide pour qui ?	22
1.2.2 Aide par qui ?	23
1.3 L'aide à la décision multicritère	24
1.3.1 Critères	25
1.3.2 Actions	26
1.4 Les Systèmes Interactifs d'Aide à la Décision	27
1.5 Théorie Multi-Attributs	30
1.6 Apprentissage automatique	31
1.6.1 Apprentissage supervisé	31
1.6.2 Apprentissage non supervisé	32
1.6.3 Apprentissage par renforcement.....	33
1.7 Conclusion.....	33
Chapitre 2 : Opérateurs d'agrégations.....	34
2.1 Qu'est-ce qu'un opérateur d'agrégation ?	34
2.2 À quoi servent les opérateurs d'agrégation?	36
2.3 Quelques opérateurs d'agrégation	37
2.3.1 La somme pondérée	37
2.3.2 La somme pondérée ordonnée.....	39
2.3.3 L'intégrale de Choquet.....	41
2.3.4 L'intégrale de Sugeno	43
2.3.5 Multi-objective optimization based on ratio analysis (MOORA).....	44
2.3.6 Complex proportional assessment of alternatives (COPRAS).....	45
2.3.7 Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)	46

2.3.8 Weighted aggregated sum product assessment (WASPAS)	48
2.3.9 Evaluation based on Distance from Average Solution (EDAS)	49
2.4 Classification des opérateurs d'agrégation.....	51
2.4.1 Opérateurs de compromis :	52
2.4.2 Agrégation conjonctive.	52
2.4.3 Agrégation disjonctive.	53
2.4.4 Agrégation mixte.....	53
2.5 Propriétés générales.....	54
2.5.1 Idempotence	54
2.5.2 Continuité	55
2.5.3 Symétrie	55
2.5.4 Associativité	55
2.5.5 Élément neutre et élément absorbant	56
2.6 Propriétés des opérateurs évoqués.....	57
2.7 Conclusion.....	58
Chapitre 3: Systèmes de recommandation	59
3.1 Collecte d'Information.....	59
3.1.1 Collecte de données explicite – Filtrage actif	59
3.1.2 Collecte de données implicite – Filtrage passif.....	60
3.2 Profil Utilisateur.....	60
3.3 Liste de recommandations.....	61
3.4 Types de système de recommandation.....	61
3.4.1 Recommandation Personnalisée.....	61
3.4.2 Recommandation par le contenu	62
3.4.3 Recommandation Sociale	66
3.4.4 Recommandation Hybride.....	71
3.5 Efficacité d'un algorithme de recommandation.....	75
3.6 Conclusion.....	76
Chapitre 4: STROMa	77
4.1 Architecture détaillée de STROMa	77
4.1.1 Problèmes de décision.....	77
4.1.2 Critères	80
4.1.3 Alternatives	81
4.1.4 Choix de l'opérateur d'agrégation.....	82

4.1.5	Caractéristiques techniques de STROMa.....	86
4.2	Les opérateurs d'agrégation mis en œuvre.....	87
4.2.1	Somme pondérée.....	88
4.2.2	Intégrale de Choquet.....	88
4.2.3	MOORA.....	92
4.2.4	COPRAS.....	93
4.2.5	EDAS.....	96
4.2.6	WASPAS.....	97
4.2.7	TOPSIS.....	99
4.3	Comment choisir un opérateur d'agrégation face à un problème de décision.....	101
4.4	Conclusion.....	105
Chapitre 5:	Expérimentations.....	106
5.1	Méthodologie.....	106
5.1.1	Outils disponibles.....	106
5.1.2	Sélection des outils d'évaluation.....	114
5.2	Expérimentations.....	115
5.2.1	MovieLens.....	115
5.2.2	Choix d'un cuisinier.....	118
5.2.3	Sélection d'un robot industriel.....	120
5.3	Conclusion.....	122
Conclusion générale	124
Références	126
Annexes	134
Table des illustrations	143
Table des tableaux	144

Introduction

Les systèmes de recommandation occupent de nos jours une place importante dans la communauté de recherche. Cette activité accrue autour des systèmes de recommandation conduit à une multiplication des méthodes. D'un point de vue pratique, cela présente des avantages mais aussi des inconvénients. Les avantages notables se trouvent au niveau de la diversité des choix des méthodes disponibles. En effet, face à un problème de décision spécifique, on aura la possibilité d'utiliser telle ou telle approche. Par contre cette multiplication des méthodes disponibles provoque une certaine difficulté au niveau du choix de l'approche. Sur quels éléments se baser pour choisir une méthode par rapport à une autre face à un problème de décision? La diversité des problèmes de décision ajoute aussi une couche de complexité supplémentaire au niveau du choix de la méthodologie. En effet, chaque problème de décision est différent et possède des caractéristiques bien propres à lui.

Par ailleurs, certains problèmes de décision se prêtent mal à une modélisation telle que conçue en recherche opérationnelle classique. En effet, la prise de conscience de la complexité des problèmes organisationnels et des limitations de la rationalité ont conduit à considérer que certains outils comme la programmation mathématique ne pouvaient plus convenir. La recherche de nouveaux outils, conjuguée au déplacement des positions épistémologiques ont donné naissance à d'autres disciplines qui s'apparentent à la recherche opérationnelle, sans pour autant totalement s'y confondre. L'aide à la décision, qui a vu le jour depuis plus d'une cinquantaine d'années, est l'une de ces disciplines.

L'essor considérable de théories telles que la théorie du choix social, la théorie de la décision combiné aux développements technologiques, l'intérêt nouveau pour l'étude des problèmes complexes, la prise en compte de certains acteurs et l'apparition de nouveaux acteurs ont renforcé les bases d'une science de l'aide à la décision. Cette science est fondée sur la conception de *«l'activité de celui qui, par des voies dites scientifiques, aide à obtenir des éléments de réponses à des questions que se posent des acteurs impliqués dans un processus de décision, éléments concourant à éclairer la décision en vue de favoriser un comportement des acteurs de nature à accroître la cohérence entre l'évolution du processus d'une part, les objectifs et/ou les systèmes de valeurs au service desquels ces acteurs se trouvent placés d'autre part.»* [1]; On parle également d'aide multicritère à la décision.

Introduction

Un gestionnaire, confronté à une situation décisionnelle, est souvent appelé à concilier plusieurs aspects ou points de vue, souvent conflictuels et non-commensurables, mal accommodés par les techniques de l'optimisation. Compte tenu de la nature multidimensionnelle des situations décisionnelles et des enjeux liés à celles-ci, il est précieux de disposer d'outils (méthodes, procédures, approches opérationnelles, etc.) fondés sur les enseignements d'une science de l'aide à la décision et permettant d'apporter une aide méthodique à la décision. Ces outils pourraient aider le gestionnaire à choisir parmi les différentes possibilités qui s'offrent à lui. L'application des modèles mathématiques au monde réel, le développement de nouvelles approches de gestion et l'importance des investissements pour des activités de conseil ont suscité des recherches ayant pour but de proposer des outils permettant de répondre à cette préoccupation. C'est ainsi que des procédures comme Somme pondérée, intégrale de Choquet, MOORA, COPRAS, TOPSIS, EDAS, WASPAS ont vu le jour. Ces procédures d'agrégation ont vu le jour car il est nécessaire d'agrèger toutes les performances pour avoir une seule valeur par alternative qui permettra de classer les alternatives. Nous détaillons dans le chapitre 2 ces différents opérateurs d'agrégation qui peuvent paraître mystérieux pour un non initié.

En dépit de l'existence d'une panoplie de méthodes d'agrégation, aucune d'entre elles n'est parfaite [2] et de toute évidence il serait illusoire de penser pouvoir développer une 'super' procédure applicable dans toutes les situations décisionnelles. Par conséquent, les différentes procédures multicritères seront utiles pour appréhender des situations décisionnelles spécifiques [3]. Toutefois, force est de constater que les différentes procédures d'agrégation multicritères, qu'elles soient proposées dans la littérature suite à des considérations théoriques ou issues d'applications réelles, ne résistent pas de la même façon à une analyse rigoureuse basée sur des préoccupations pragmatiques ou théoriques. Le problème est alors de savoir comment choisir l'opérateur d'agrégation qui se prête le mieux à une situation décisionnelle particulière? Cette question occupe une place importante dans ces travaux de thèse. En effet, l'objectif de cette thèse est d'offrir aux décideurs un système de recommandation mettant en œuvre plusieurs opérateurs d'agrégation et de l'assister dans le choix d'un opérateur face à un problème de décision.

Cette thèse se décompose en 5 chapitres, qui sont :

Premier chapitre, dans ce chapitre nous présentons les systèmes d'aide à la décision. Notamment l'avènement de l'aide à la décision à nos jours. Il est question également des

Introduction

systèmes d'aide à la décision multicritère notamment les deux écoles qui émergent en analyse multicritère à savoir l'école européenne et américaine. Il est aussi question des systèmes interactifs d'aide à la décision dont le principe consiste à trouver la bonne solution grâce aux préférences de l'utilisateur au fur et à mesure de l'utilisation du système. Et enfin nous parlons des méthodes d'apprentissage dans les systèmes d'aides à la décision.

Le deuxième chapitre porte sur les opérateurs d'agrégation. Nous introduisons la notion d'opérateur d'agrégation, ensuite nous verrons à quoi servent les opérateurs d'agrégation dans les systèmes d'aide à la décision. Par la suite différents opérateurs d'agrégation sont discutés tels que la somme pondérée, l'intégrale de Choquet, l'intégrale de Sugeno, MOORA, COPRAS, etc. Il est aussi question de la classification des opérateurs d'agrégation ainsi que leurs propriétés.

Le troisième chapitre présente les systèmes de recommandations. Les systèmes de recommandation sont des systèmes d'aide à la décision permettant de fournir à l'utilisateur des éléments susceptibles de l'intéresser afin de l'aider dans sa prise de décision. Nous introduisons différents types de système de recommandation à savoir la recommandation personnalisée qui se base sur le comportement passé de l'utilisateur, la recommandation par le contenu qui exploite les propriétés intrinsèques des objets, la recommandation sociale ou collaborative qui utilise l'historique d'autres utilisateurs afin d'émettre des recommandations, et enfin la recommandation hybride qui combine ces trois précédentes approches.

Le quatrième chapitre donne des détails sur le système de recommandation mis en œuvre à savoir STROMa. Dans ce chapitre, nous présentons différents composants de STROMa à savoir les problèmes de décision, les alternatives et les critères. Il est aussi question du choix d'un opérateur d'agrégation. Ce choix peut être manuel ou automatique. Le choix manuel consiste à ce que l'utilisateur sélectionne lui-même un opérateur d'agrégation dans une procédure d'agrégation. Quant au choix automatique, il s'agit d'une proposition d'opérateur d'agrégation par le système par rapport au problème de décision de l'utilisateur. Nous détaillons également dans ce chapitre l'algorithme des différents opérateurs d'agrégation mis en œuvre.

Le cinquième et dernier chapitre concerne les expérimentations que nous avons réalisées dans le but de tester les performances de notre système. Nous détaillons dans un premier temps les méthodologies de test. Il s'agit des méthodes permettant de mesurer la

Introduction

capacité qu'a un système de recommandation d'émettre de bonnes recommandations. Nous distinguons trois catégories de mesures à savoir : les mesures de précision de la prédiction de note, les mesures de précision d'utilisation et enfin les mesures d'ordre. Ensuite nous décrivons les différents tests qui ont été effectués sur le système. Dans un premier temps, nous comparons les performances de notre système avec celui de MovieLens qui est un site de distribution de film en ligne. Puis nous testons notre système sur deux exemples classiques de la littérature à savoir le choix d'un cuisinier et la sélection d'un robot industriel. Nous terminons cette thèse bien évidemment sur une conclusion et des perspectives à ces travaux de recherche.

Rapport-Gratuit.com

Chapitre 1 : Aide à la décision Multicritère / Multiattribut

L'aide à la décision est un ensemble de concepts dont l'objectif est d'aider un utilisateur dans un processus de décision. Cette aide fournie à l'utilisateur se veut être la plus en adéquation avec ses préférences. Ainsi, pour un système d'aide à la décision, il s'agit dans un premier temps, d'apprendre les préférences de l'utilisateur pour un problème de décision. La deuxième étape consiste à faire la synthèse des informations apprises dans le but de faire des recommandations à l'utilisateur. Cette synthèse se fait à l'aide d'un opérateur d'agrégation. A cet effet, Il existe de nos jours, de multitudes d'opérateur d'agrégation que nous détaillerons dans le chapitre II. Le choix de l'opérateur d'agrégation est donc crucial dans le processus d'aide à la décision.

Ce chapitre retrace l'historique de l'aide à la décision, ensuite présente les systèmes d'aide à la décision de manière générale, puis s'intéresse aux systèmes d'aide à la décision multicritère, notamment aux deux écoles qui émergent en analyse multicritère à savoir l'école européenne et l'école américaine. Pour les systèmes interactifs d'aide à la décision le principe de fonctionnement consiste à trouver la bonne solution grâce aux préférences de l'utilisateur au fur et à mesure de l'utilisation du système. Et enfin nous aborderons les méthodes d'apprentissage dans les systèmes d'aides à la décision.

1.1. Historique de l'aide à la décision

L'origine de l'aide à la décision comme détaillée par Alexis Tsoukiàs dans [4], peut être située dans la période d'avant deuxième guerre mondiale; elle se trouve dans les études menées par l'armée britannique dans le cadre de l'installation des systèmes radar et les efforts de décodage du code secret des communications allemandes (1936-37). On trouve de plus amples explications dans [5] et [6] à ce sujet. On y trouve la première utilisation du terme "recherche opérationnelle". En réalité, les problèmes liés à la prise de décision par les personnes, les organisations, les institutions avaient déjà occupé les scientifiques et les gestionnaires à partir du XVIIIème siècle, voir [7] sur les problèmes combinatoires, [8] et [9] sur les procédures de décision collective. On retrouve également au début du XXème siècle des traces dans [10] sur les problèmes économiques à dimensions multiples, dans [11] et [12] sur la gestion scientifique des entreprises et dans [13] sur la décidabilité. Un problème de décision est dit décidable s'il existe un algorithme, une procédure mécanique qui termine en

un nombre fini d'étapes, qui le décide, c'est-à-dire qui réponde par oui ou par non à la question posée par le problème. S'il n'existe pas de tels algorithmes, le problème est dit indécidable. Dans ces contributions le concept de décision est au cœur des préoccupations.

C'est le succès de la recherche opérationnelle à organiser les activités militaires des alliés qui accrédi tera l'idée que la prise de décision est un phénomène qui peut être étudié de façon scientifique et que des modèles généraux sont concevables. Entre la fin des années 40 et le début des années 50, plusieurs contributions fondamentales vont voir le jour (voir [14], [15] pour la programmation linéaire, [16], [17], [18] pour la théorie de la décision et la théorie des jeux, [19] sur l'algorithmique et l'existence de machines capables de résoudre "tout problème"). C'est à cette époque que vont apparaître les premières sociétés savantes de recherche opérationnelle (en 1948 en Angleterre, en 1950 aux États Unis) et les premières revues du domaine [20]. Toujours à cette époque, s'effectueront les premiers travaux d'application de la nouvelle discipline dans des cas pratiques (voir [21]) ainsi que l'apparition des premières sociétés privées spécialisées en "aide à la décision" (mais le terme n'est pas encore utilisé à l'époque). Le cas le plus célèbre est la société RAND dans laquelle la recherche opérationnelle se développera en science appliquée aux problèmes de la nouvelle industrialisation de l'après-guerre. Ces premiers travaux et expériences sont caractérisés par la recherche de structures formelles sous-jacentes aux problèmes concrets et l'utilisation des mathématiques et de la logique comme langage de modélisation. Les premiers pas dans cette direction vont renforcer l'idée que des problèmes complexes de décision sont modélisables à travers l'imposition d'un modèle de rationalité simple (la maximisation d'une fonction d'utilité exprimant les préférences du décideur; l'optimisation de cette fonction apportant la solution du problème de décision). Les travaux de Von Neumann et Morgenstern [18] et de Nash [17] montrent exactement sous quelles conditions ces fonctions et ces solutions existent. D'autre part, l'algorithme de programmation linéaire développé par Dantzig (la célèbre méthode du Simplexe) [14] apportera les outils nécessaires pour la résolution de problèmes, même de grande taille. Turing [19], mais aussi Wiener [22] et Ashby [23], vont aller plus loin pour proposer une théorie de résolution générale des problèmes.

Néanmoins à cette même période commencent à apparaître des travaux qui critiquent cet édifice et son paradigme associé. En 1951, Arrow [24] publie son célèbre théorème d'impossibilité montrant que l'agrégation des préférences d'individus rationnels, sous des conditions réputées naturelles (universalité, indépendance, respect de l'unanimité, non

existence de dictateur), est impossible (si le résultat doit être un ordre complet). Ce théorème, qui, d'une part, va conclure la discussion ouverte avec les contributions de Borda et Condorcet au XVIIIème siècle (au sens que nous savons maintenant qu'il n'y a pas de procédure d'agrégation des préférences universelle, voir [25], [26]), va ouvrir la voie à une littérature immense sur la théorie du choix social (voir [27] [28] [29], [30], [31]). En 1953, ALLAIS [32] présente son célèbre paradoxe qui montre comment les axiomes, introduits par von Neumann et Morgenstern comme conditions nécessaires pour l'existence d'une fonction d'utilité (et implicitement pour exhiber un comportement rationnel), sont systématiquement violées par le comportement réel de décideurs confrontés à des choix très simples (voir aussi [33], [34]) Cette falsification empirique de la théorie de l'utilité espérée ouvre une autre voie de recherche encore, à la rencontre des études des sciences cognitives (voir par exemple [32]).

En 1947 déjà, Simon [35] commence à observer les processus de décision dans les organisations et remarque que le comportement réel de décideurs est loin d'être représentable par les postulats de la théorie de la décision, au moins dans la forme qu'elle a à ce moment-là. Pendant les années '50, Simon ([36] [37] [38]) développera sa théorie de la "rationalité limitée" selon laquelle un décideur confronté à un problème de choix se comportera selon un critère de "satisfaction", au sens de choisir la première solution qui "satisfait" ces nécessités, et non pas selon un critère d'optimisation idéal, irréaliste et inutilisable. En effet Simon considère que la théorie de la décision fait trois hypothèses implicites (voir la présentation dans [39]):

- les décideurs connaissent bien leur problème;
- ce problème est toujours représentable comme un problème de recherche d'efficacité;
- l'information et les ressources nécessaires pour trouver une solution sont toujours disponibles.

Aucune de ces hypothèses n'est validée dans la vie réelle (selon Simon):

- les décideurs n'ont jamais une idée très claires de leur problème;
- souvent les problèmes de décision se présentent comme la recherche d'un compromis;
- la solution d'un problème est soumise à des contraintes temporelles et de ressources disponibles.

L'innovation introduite par Simon est capitale. Les approches de la théorie de la décision développées jusqu'à ce moment se basent sur l'hypothèse que le modèle de rationalité existe indépendamment du décideur et du processus de décision. Simon mettra au centre de sa réflexion le processus de décision (les activités mentales qu'un décideur doit effectuer pour prendre une décision) et postulera que le modèle de rationalité doit être recherché dans ce processus et non en dehors. Une hypothèse de "rationalité externe" est compatible avec un modèle d'optimisation. Cela ne pose aucun problème ; il n'en va pas de même dans l'hypothèse d'un modèle défini de façon subjective. Les travaux de Simon vont ouvrir plusieurs pistes de recherche orientées à la fois vers la création de nouvelles approches d'aide à la décision (voir par exemple [40] et vers ce qui ensuite sera connu sous le nom d'"intelligence artificielle" (voir [41]). Il faut remarquer en particulier que l'idée de recherche d'une solution satisfaisante va trouver une correspondance immédiate dans le problème de trouver un compromis acceptable dans une décision en présence de critères multiples (voir [42]). Entre la fin des années '50 et le début des années '70 vont apparaître toute une série de méthodes classiques de la nouvelle discipline, ouvrages sur lesquels plusieurs générations de chercheurs et de professionnels vont se former ([43] [44], [45], [46], [47], [48]). Les années '50 et '60 vont connaître la multiplication des recherches, des cours universitaires, des applications dans des domaines très différents. Les grands clients des études de recherche opérationnelle et d'aide à la décision sont les entreprises qui administrent des réseaux (distributeurs d'eau, télécommunications, chemins de fer, compagnies aériennes). Par ailleurs, de nombreuses entreprises spécialisées en aide à la décision vont apparaître (en France, signalons la création de la SEMA-METRA, la revue scientifique de cette société étant devenue une des plus importantes revues de la discipline). D'autre part, il ne faut pas oublier que ces années sont celles de la reconstruction après la guerre; elles vont voir la mise en mouvement de formidables ressources pour la résolution de problèmes formidables eux aussi. Sans surprise, c'est pendant cette période que commenceront à se développer les approches critique du paradigme dominant de la théorie de la décision (pour les premières discussions voir [49], [50]). Ces critiques vont se développer dans plusieurs directions.

Au début des années 60 paraît le célèbre papier de Zadeh ([51]) sur les ensembles flous; celui-ci va introduire une nouvelle perspective dans le traitement de l'incertitude, de l'ambiguïté et des variables linguistiques. L'innovation de Zadeh aura un impact majeur dans le futur de la discipline parce qu'elle concerne une partie fondamentale du langage formel: la théorie des ensembles. L'extension du concept d'ensemble avec l'introduction du concept de

“fonction d’appartenance”, une “mesure” de l’appartenance d’un élément à un ensemble, va permettre d’augmenter l’expressivité et la flexibilité des langages formels et par conséquent des modèles d’aide à la décision.

Un autre domaine qui va apporter des contributions majeures au développement de la théorie de la décision et aux approches alternatives est celui des sciences cognitives et de la psychologie (voir [52], [53] [54], [55]). L’intuition d’Allais dans les années ’50 de valider expérimentalement les axiomes de la théorie de la décision a donné suite à plusieurs travaux dans la même lignée. Nous voulons signaler ici en particulier les travaux de Tversky (voir [56], [57], [58]) qui vont mettre en évidence le fait que les idées intuitives sur les propriétés des relations de préférence sont plus une exigence de la théorie qu’un comportement des décideurs réels. Tversky démontrera que la relation de préférence peut bien exhiber des intransitivités ([57]) et que la relation d’indifférence (ou de similarité) peut bien ne pas être symétrique ([58]). Ces travaux mettent clairement en évidence la nécessité d’une étude plus approfondie des structures de base utilisées dans la construction des modèles d’aide à la décision, notamment les structures des relations de préférence ([59], [60], [61]) et les fonctions qui doivent “mesurer” la préférence (les fonctions de valeur ou d’utilité, voir [62], [63]). Pour voir du travail plus récent le lecteur consultera [64], [65], [66], [67] (voir encore la récente revue dans [68]). Toujours dans le cadre de l’influence des études psychologiques il ne faut pas négliger l’apparition dans les années ’60 du courant de la psychologie “relationnelle” fondée sur une approche définie par ces auteurs comme “constructive” (voir [69], [70], [71]). Dans cette approche sera mise en évidence l’importance de la formulation des problèmes et de la relation entre celui qui demande une aide et celui qui l’apporte (le patient et le thérapeute dans leur terminologie). Surtout cette approche mettra en évidence le fait que les problèmes ne sont pas les données d’un processus de décision: le processus pour définir et pour résoudre un problème est le même. Dans cette perspective, la solution est une construction et non pas le résultat d’une recherche dans un espace donné. Nous pouvons conclure l’histoire en remarquant qu’à la fin des années ’60, début des années ’70; la recherche opérationnelle et la théorie de la décision se trouvent à un point de fort développement tant du point de vue scientifique que professionnel.

1.2. Aide à la décision

L'aide à la décision est un ensemble de méthodes et de techniques, permettant à une personne confrontée à un problème de décision, de l'aider à prendre une décision. Elle est utilisée dans différents domaines tels que la finance [72], la gestion [73], la médecine [74], etc.

L'activité d'aide à la décision est une activité faite par tous presque chaque jour. Nous prenons des décisions sans arrêt. Des plus simples: "Je prends mon parapluie ou non?" [75], aux plus complexes "Comment appliquer l'accord sur le désarmement?" [76]. À tous les niveaux: individuel "Je divorce?" [77], organisationnel "comment gérer les roulements du personnel ?" [78], inter-organisationnel "Par où faire passer l'autoroute?" [79]. Assez souvent, dans ces processus de décision nous demandons l'avis, l'aide, le support d'ami(e)s, d'experts, de sociétés de conseil, etc. Nous pouvons nous poser les questions suivantes: est-ce que qu'on peut concevoir une aide à la décision indépendante du domaine, adaptable à toute situation? Existe-il un expert en aide à la décision qui ne soit pas expert d'un domaine? Quelle est la différence entre un expert de ce type et un psychologue, un médecin, un avocat, un expert en productique ou encore un(e) ami(e) de cœur?

Les méthodes d'aide à la décision permettent de fournir des informations sur le problème (phase d'intelligence), des outils pour l'analyser ainsi que pour trier les informations pertinentes qui peuvent aider le décideur (phase de conception) dans le but de formuler plusieurs alternatives qui pourront être proposées à ce décideur (phase de choix). Ces étapes se retrouvent dans toutes les prises de décision que Simon a définies comme étant un processus cognitif [80]. Nous pouvons retrouver plusieurs types d'acteurs lors d'une prise de décision. La personne qui va prendre la décision est le décideur. C'est à lui que s'adresse ce processus d'aide à la décision dans lequel il est impliqué pour exprimer ses préférences. Un autre acteur important de ce processus est l'homme d'étude [81]. C'est un expert qui va aider le décideur à comprendre le problème, lui donner des informations essentielles et lui expliquer les conséquences des différentes alternatives lorsque c'est possible. Généralement ces deux acteurs sont deux personnes distinctes. D'autres catégories d'acteurs peuvent intervenir pendant le processus comme le demandeur qui est à l'origine de la requête d'aide à la

décision, les intervenants¹ qui vont donner leurs avis en fonction de leurs préférences ou les agis² qui regroupent toutes les personnes concernées par les conséquences de la décision.

Dans [81], Bernard Roy nous propose la définition suivante de l'aide à la décision :

Définition. *L'aide à la décision est l'activité de celui qui, prenant appui sur des modèles clairement explicités mais non nécessairement complètement formalisés, aide à obtenir des éléments de réponses aux questions que se pose un intervenant dans un processus de décision, éléments concourant à éclairer la décision et normalement à prescrire, ou simplement à favoriser, un comportement de nature à accroître la cohérence entre l'évolution du processus d'une part, les objectifs et le système de valeurs au service desquels cet intervenant se trouve placé d'autre part.*

L'objectif final de l'aide à la décision n'est donc pas de prendre la décision à la place du décideur mais bien de lui donner les informations pertinentes pour qu'il puisse les comprendre, lui donner des éléments de solution et les conséquences de chaque alternative afin qu'il puisse prendre la décision la plus appropriée. Il ne s'agit pas de décision optimale mais de décision satisfaisante. En effet Simon explique dans son principe de la rationalité limitée [67] que certaines hypothèses admises à l'époque concernant la rationalité de certaines décisions n'étaient pas valides. Il redéfinit la rationalité classique en rationalité limitée et énonce plusieurs hypothèses :

- L'accès à l'information est limité. Avec les nouvelles technologies et l'accès à internet, on pourrait croire que l'accès à l'information est illimité. Mais s'informer demande beaucoup de ressources que ce soit en termes d'argent ou de temps. Il n'est donc pas possible d'obtenir toutes les informations disponibles pour un problème donné.
- La capacité cognitive d'un individu est elle aussi limitée. Il n'est pas possible pour une personne d'analyser toutes les informations disponibles, de comprendre toutes les situations possibles et d'optimiser son choix. Lorsqu'un décideur arrive à trouver une solution satisfaisante, il doit s'en contenter et ne pas chercher la

¹ Les acteurs (individus, corps constitués, collectivités, etc.) sont ceux que nous appelons les intervenants dans la mesure où, de par leur intervention, conditionnent directement la décision en fonction du système de valeurs dont ils sont porteurs.

² Les administrés, les contribuables, personnes âgées, personnel d'une entreprise, étudiants d'une université, consommateurs, etc. qui de façon normalement passive, subissent les conséquences de la décision, laquelle est seulement censée tenir compte de leur préférences, sont appelés les agis.

solution optimale, au sens mathématique, car il arrive que cette solution n'existe pas.

- Le décideur a une vision floue de ses préférences et n'a pas une idée claire de son problème. Les préférences d'un décideur peuvent évoluer pendant le processus de décision et les critères utilisés pour prendre une décision peuvent être contradictoires. Il n'est alors pas possible de trouver une décision optimale. Prenons l'exemple d'une personne voulant acheter une voiture. La voiture doit être très puissante mais aussi la plus écologique possible. Un moteur puissant générera toujours plus de pollution qu'un moteur plus modeste, il est alors impossible de maximiser les deux critères. Le décideur devra faire un compromis entre les deux critères : puissance et écologie.

L'information est une donnée essentielle de l'aide à la décision. Ces dernières années, le développement des outils et du matériel informatique a permis à de nombreux systèmes d'aide à la décision de voir le jour. Il est aujourd'hui possible, en utilisant des systèmes d'aide à la décision, de traiter beaucoup plus d'informations qu'une personne seule.

1.2.1. Aide pour qui ?

Elaborer un système d'aide à la décision passe par identifier à qui cette aide est destinée. Selon Bernard ROY dans [81], l'aide s'adresse généralement à l'un des intervenants du processus de décision. Or, ces intervenants peuvent être relativement variés, poursuivre des objectifs différents, avoir des systèmes de valeurs conflictuels. Il s'ensuit que l'aide à la décision ne peut que rarement s'exercer globalement et sur la base d'un unique modèle au profit de tous les intervenants du processus. C'est pourquoi faire de l'aide à la décision suppose, le plus souvent, qu'un intervenant particulier (individu, corps constitué ou collectivité) ait été identifié. Nous l'appellerons le décideur.

Identifier un décideur, c'est essentiellement préciser les objectifs au service desquels il est normalement placé. Cet intervenant particulier joue généralement un rôle déterminant dans la conduite du processus. C'est pour son compte ou en son nom que l'aide à la décision s'exerce. Mais cet intervenant particulier peut aussi être le porte-parole des agis même si l'intervention de ce dernier dans l'évolution du processus de décision n'apparaît pas a priori comme déterminante.

Lorsque l'aide à la décision peut être moins personnalisée du fait d'un consensus suffisant entre tout ou partie des intervenants et agis, nous continuerons à parler de décideur. Celui-ci apparaît comme une entité un peu mythique, définie par les objectifs communs ou présumés tels. Si, comme c'est fréquemment le cas, il faut pour les besoins de la modélisation, préciser davantage le système de valeurs, des difficultés peuvent surgir. Le « bon » et le « mauvais », n'ont qu'exceptionnellement un sens absolu et prêtent vite à discussion. Celle-ci peut être sans issue si elle se réfère à un décideur trop vague.

En conclusion, le décideur désigne en dernier ressort l'entité qui apprécie le « possible » et les finalités, exprime les préférences et est sensé les faire prévaloir dans l'évolution du processus. Aider ce décideur n'implique pas pour autant que seuls ses opinions, stratégies ou préférences sont à modéliser à l'exclusion de celles des autres intervenants.

Notre étude s'intéresse aussi à l'homme d'étude, qui, vu les multitudes de méthodes existantes, se trouve au cœur d'un dilemme qu'est le choix de la méthode à choisir.

1.2.2 Aide par qui ?

Dans le cas où un responsable précis incarne le décideur, il peut exercer lui-même l'activité d'aide à la décision. Mais le décideur est, à l'égard de ses propres problèmes, dans une situation qui n'est pas sans ressemblance avec celle du médecin vis-à-vis de sa propre santé. De plus, les compétences nécessaires peuvent lui faire défaut. C'est pourquoi celui qui prend en charge l'aide à la décision est généralement distinct du décideur. Nous l'appellerons, dans tous les cas, l'homme d'étude.

C'est à dessein que nous employons ce mot et non celui d'analyste plus usité. Ce dernier, en effet, conduit à accrédi-ter l'idée selon laquelle l'homme d'étude (que l'on pourrait encore appeler le rapporteur) est totalement extérieur au problème et au processus de décision qu'il analyse sans les influencer en quoi que ce soit.

L'homme d'étude est habituellement un expert ou un spécialiste (chercheur opérationnel, économiste, statisticien, financier, ingénieur,...) travaillant seul ou ayant la responsabilité d'une équipe. Il peut être un proche collaborateur du décideur, ou quelqu'un qui lui est totalement étranger (provenant d'un service d'étude de l'état, d'un département fonctionnel d'une entreprise ou encore d'un bureau d'étude spécialisé).

Le rôle de l'homme d'étude consiste entre autres à expliciter le modèle, à l'exploiter en vue d'obtenir des éléments de réponses, à éclairer le décideur sur les conséquences de tel ou tel comportement en les lui rendant intelligibles, éventuellement en prescrivant (préconisant, conseillant) une série d'action ou encore une méthodologie. Le succès de l'homme d'étude dépend beaucoup de la manière dont, dans le cadre des moyens qui lui sont alloués, il délimite le modèle, affine la problématique, contrôle les données et choisit l'approche opérationnelle.

1.3 L'aide à la décision multicritère

L'aide à la décision multicritère MCDA [81] est un sous domaine de l'aide à la décision, dont les différentes alternatives ou actions, sont évaluées à travers plusieurs critères. Dans ces problèmes de décision, l'utilisation d'un seul critère ne permet pas de départager efficacement les différentes alternatives tout en tenant compte de l'ensemble des préférences du décideur.

Il existe un très grand nombre de méthodes d'aide à la décision multicritère. Deux « écoles » se distinguent dans le domaine MCDA et suivent des principes de base assez différents [82]. La première est « l'école Américaine », qui utilise le plus souvent une fonction d'utilité additive qui permet de combiner les valeurs d'utilité en une note globale pour l'action. La méthode la plus simple de cette catégorie est celle de la somme pondérée, où la note globale d'une action est la somme pondérée des notes de chaque critère de sélection multipliée par le poids du critère. Les méthodes de ce type sont MAUT, Multi Attribute Utility Theory [83], MHM pour Méthode de hiérarchisation multicritère [84] ou AHP pour Analytical Hierarchy Process. » [85].

L'école Européenne favorise en général l'utilisation de méthodes fondées sur des notions de surclassement entre les actions potentielles. La notion de surclassement correspond à dire qu'un élément est préféré à un autre par rapport à un ou plusieurs points de vue.

« Les méthodes les mieux connues sont les méthodes ELECTRE, ÉLImination Et Choix Traduisant la RÉalité [86] [87] et PROMETHEE, Preference Ranking Organisation METHod for Enrichment Evaluations [88] . Les relations de surclassement permettent de construire le graphe de surclassement, à partir duquel est construit le classement final des actions potentielles. Ces méthodes permettent aussi d'établir des notions d'indifférence et

d'incomparabilité entre les actions, ce qui correspond souvent le mieux à la façon de penser du décideur. Malheureusement ces deux notions soulèvent souvent des problèmes d'interprétation du graphe de surclassement. De nombreuses modifications aux méthodes de base ont été proposées, notamment pour les méthodes ELECTRE, ce qui en explique le grand nombre de variantes. Ces modifications sont liées à l'introduction de plusieurs seuils de surclassement et à l'utilisation de concepts flous. Ces modifications peuvent éviter la définition de fonctions d'utilité pour les critères de sélection, tout en permettant d'améliorer les performances techniques de la méthode. La méthode devient néanmoins souvent encore plus difficilement accessible au décideur. » [85]

Il existe encore un grand nombre d'autres méthodes qui n'appartiennent ni à l'une ni à l'autre des deux « écoles », comme QUALIFLEX [89]. Cette méthode tient compte du fait qu'il est davantage dans la nature humaine de donner un rang à des objets que de leur attribuer des côtes. Il y a également, par exemple, les méthodes utilisant des principes de coût et bénéfice [90] [91] et [92] [93]. Toutes ces méthodes demandent, elles aussi, l'obtention a priori de poids et d'autres paramètres.

À présent que le cadre multicritère est posé, il nous faut définir les différents types de critères existants.

1.3.1 Critères

Un critère est une fonction $g: A \rightarrow R$ qui permet, relativement à un point de vue donné et pour un acteur identifié, de comparer deux actions a et b .

$$g(a) \geq g(b) \rightarrow aSb$$

Où 'S' signifie surclasse : « est au moins aussi bon que » (cela suppose que g est un critère à maximiser)

La manière dont s'effectue la comparaison des actions sur chaque critère dépend du pouvoir de discrimination de celui-ci [94], [95] [96]. Lorsque le pouvoir de discrimination de g est parfait, on dit que g est un vrai-critère. Mais si g possède un pouvoir de discrimination imparfait, il s'agira d'un quasi-critère ou d'un pseudo-critère.

Un critère est souvent utilisé de la façon suivante :

$$g(a) > g(b) \leftrightarrow aPb$$

$$g(a) = g(b) \leftrightarrow aIb$$

P : indique une préférence stricte, tandis que **I** signifie une indifférence. Pour nuancer la transition entre I et P, on introduit une zone intermédiaire appelée préférence faible et notée Q.

Ainsi g induit sur A un pré-ordre total, structure adéquate pour déterminer un choix, un tri ou un classement. En effet, g permet de pouvoir comparer chaque élément 2 à 2, l'un par rapport l'autre, et permet ainsi de connaître lequel est préféré, et ce pour chaque couple d'actions.

1.3.2 Actions

D'après Bernard Roy dans [81], une action « A » est la représentation d'une éventuelle contribution à la décision globale susceptible, eu égard à l'état d'avancement du processus de décision, d'être envisagée de façon autonome et de servir de point d'application à l'aide à la décision (ce point d'application pouvant suffire à caractériser « A »).

Cette définition appelle une explication (à propos du terme « autonome ») ainsi que quelques remarques avant même de pouvoir être illustrée à partir d'un premier exemple de référence.

Chaque action « a » introduite dans un modèle doit avoir un sens par elle-même. Dire qu'elle représente une contribution autonome (à la décision finale) signifie qu'elle peut être considérée isolément de toute autre action sans pour autant perdre sa portée décisionnelle ou sa valeur de point d'application de l'aide à la décision. Par exemple, pour un projet d'autoroute, dont il s'agit de préciser le tracé permet d'envisager diverses variantes géographiquement éloignées, sans parties communes, des éventualités telles que : faire un tunnel, une tranchée couverte ou une simple tranchée ne présentent pas une autonomie suffisante pour définir des actions. En revanche, si le projet implique que tous les trajets traversent un site critique, chaque façon de le traverser peut donner naissance à une action.

Pour la clarté du travail d'aide à la décision, il est utile d'introduire les distinctions suivantes :

- aux actions réelles issues d'un projet complètement élaboré susceptible d'être mis à exécution s'opposent des actions plus ou moins fictives, ce qualificatif de fictif servant à souligner le fait que l'action correspond à un projet idéalisé, incomplètement élaboré ou encore construit dans l'imagination ;
- aux actions réalistes correspondant à un projet dont la mise à exécution peut être raisonnablement envisagée s'opposent des actions plus ou moins irréalistes : celles-ci peuvent, par exemple, correspondre à la satisfaction d'objectifs incompatibles tout en constituant un bon support de discussion et de raisonnement.

Il est clair qu'une action fictive peut être réaliste ou non réaliste.

Une alternative, souvent appelée action, représente une solution parmi tant d'autres dont dispose le décideur face à un problème de décision. C'est dans cet ensemble d'alternatives que la recherche de solution la plus adaptée au problème du décideur s'effectue.

Chaque alternative est évaluée sur chaque critère par le décideur, formant ce qu'on appelle la matrice de performance.

1.4 Les Systèmes Interactifs d'Aide à la Décision

Dans [97] VINCKE définit une méthode interactive d'aide à la décision multicritère comme suit :

Définition : Une méthode interactive consiste en une alternance d'étapes de calculs et d'étapes de dialogue avec le décideur. La première étape de calculs fournit une première solution. Celle-ci est présentée au décideur qui réagit en apportant des informations supplémentaires sur ses préférences (étape de dialogue). Cette information est injectée dans le modèle utilisé et permet de construire une nouvelle solution.

Le dialogue signifie dans ce cas une contribution active du décideur à la construction de la solution au lieu de définir a priori les paramètres du calcul utilisé, comme c'est le cas dans les systèmes d'analyse de décision multicritère (MCDA). Ainsi le système continue d'apprendre les préférences du décideur au fur et à mesure de l'utilisation du système. Cela permet de peaufiner et surtout d'actualiser les connaissances apprises sur le décideur au cours du temps.

Pour ce faire, le processus utilisé est un processus interactif. Dans lequel le système associé présente à chaque étape ses recommandations au décideur. Celui-ci estime si ce résultat qui lui est présenté est satisfaisant ou non à partir des informations fournies. Dans le cas où il n'est pas satisfait, il peut continuer à introduire de l'information sur ses préférences, en continuant à utiliser le système ou en fournissant des notes sur des éléments proposés afin de trouver des solutions plus satisfaisantes. Cette procédure peut être illustrée par le schéma suivant extrait des travaux de DONG dans [98].

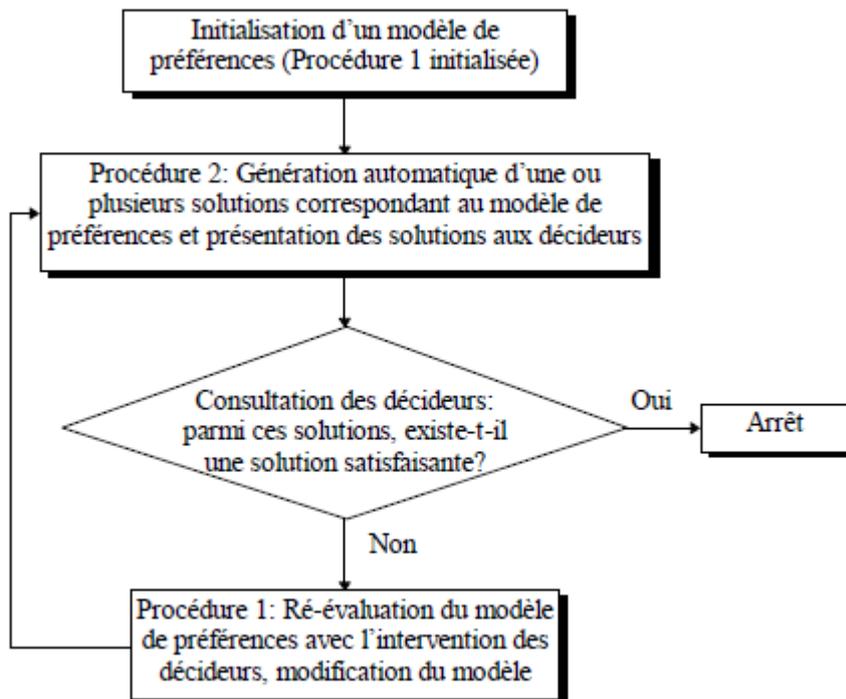


Figure 1 : Processus interactif de décision

L'objectif d'une telle méthodologie interactive n'est pas d'identifier un modèle de préférence préexistant dans la tête du décideur, mais de le faire participer en ajoutant des informations dans le système jusqu'à ce qu'il soit satisfait.

« L'interactivité permet également au décideur de mieux comprendre ses préférences concernant le problème traité et de les modifier en fonction d'une compréhension progressivement améliorée. Cette approche facilite en même temps l'acceptation des recommandations du système, puisque le décideur peut continuer le dialogue jusqu'à ce qu'il soit convaincu que le modèle de préférence est bien le sien. » [85]

Ce fonctionnement permet de mettre en évidence que les utilisateurs ne sont pas, la plupart du temps, objectifs vis-à-vis de leurs préférences. Cette méthodologie leur permet de corriger la vision qu'ils ont par rapport à ce qui est important pour eux. Ce fonctionnement implique que le système s'améliore à chaque information reçue et diminue le risque de bruit dans les informations.

« Les modifications dans les préférences du décideur font partie de la rationalité limitée générale des décideurs, qui est un aspect bien évoqué dans [99] et davantage approfondi dans [100]. Pour cette raison, il est impossible ou non souhaitable d'essayer de déterminer automatiquement si une recommandation est vraiment satisfaisante pour le décideur ou non. Simon constate que seul le décideur lui-même peut déterminer son niveau de satisfaction concernant une recommandation. C'est pour cette raison qu'une approche d'optimisation classique n'est pas appropriée pour résoudre les problèmes de décision multicritère. » [85]

L'utilisation du principe d'interactivité permet ainsi à l'utilisateur de déterminer de lui-même quand il est satisfait des éléments qui lui sont proposés. La rationalité limitée des utilisateurs nécessite d'être prise en compte à travers des systèmes permettant à l'utilisateur de fournir des informations contradictoires. C'est le cas lorsqu'il change de préférences, il faut donc modéliser les préférences de l'utilisateur et avoir la capacité de mettre à jour ce modèle rapidement, et cela sans être bloqué par les informations précédemment fournies (comme le serait un système à base de règles). C'est-à-dire qu'il faut prendre en compte les informations précédemment obtenues sur l'utilisateur. En cas de contradiction entre les anciennes informations et les nouvelles informations, il faut utiliser les informations les plus récentes sans pour autant ne pas oublier les anciennes. Il faut ainsi privilégier les nouvelles informations en utilisant par exemple une distribution de gradient.

Un regroupement peut être fait entre le domaine de l'aide à la décision et les méthodes d'agrégation. Ce regroupement est appelé Procédures d'Agrégation Multicritère (PAMC) voir [101].

Le système d'aide à la décision STROMa(SysTem of RecOmmendation Multi-criteria) [102], que nous avons mis en place, utilise ce processus interactif d'aide décrit plus haut. Nous verrons que l'utilisateur est amené à noter l'opérateur d'agrégation proposé par le

système par rapport à un problème de décision. En fonction de cette note, l'opérateur est renforcé ou affaibli selon le degré de satisfaction du décideur.

Afin de modéliser numériquement les préférences d'un utilisateur, la théorie multi-attribut a été développée.

1.5 Théorie Multi-Attributs

Les premières tentatives d'aide à la décision, avec objectifs multiples ou non, d'après Martin Arnaud dans [103], remontent à la fin des années 1960 à travers les travaux de Raïffa et Edwards [104] [105] qui ont donné naissance à l'analyse de la décision (Decision Analysis). Dans ces travaux, le modèle formel de référence fait appel à la représentation numérique des préférences du décideur sur l'ensemble des choix possibles grâce à une fonction de préférence dite fonction d'utilité. L'idée de base d'une telle démarche étant que l'encodage (c'est-à-dire la construction) d'une fonction d'utilité dans un contexte de décision donné permettra d'affecter des "scores" ou utilités, aux actions potentielles (c'est-à-dire les choix possibles) auxquelles fait face le décideur. Ces scores permettront ensuite de classer les actions de la moins désirable à la plus désirable (et vice-versa). [106].

On parle de fonction d'utilité multi-attribut vis-à-vis d'une "combinaison simple" de fonctions d'utilité mono-objectif. Ainsi l'objectif de la théorie de l'utilité multi-attribut est de modéliser numériquement les préférences d'un décideur.

Le concept d'utilité est relativement simple : il s'agit d'utiliser une fonction $U : X \rightarrow \mathbb{R}$ attribuant à tout objet de l'ensemble de choix un nombre réel de telle sorte que plus ce dernier est élevé, plus l'objet est "préférée" par le décideur.

Soit $x := (x_1, \dots, x_n)$, avec $n \geq 2$, un ensemble d'objets d'intérêt décrits par un ensemble $\mathbb{N} = \{1, \dots, n\}$ d'attributs.

Ces préférences prennent mathématiquement la forme d'une relation binaire qu'il s'agit de représenter par le biais d'une fonction d'utilité globale telle que :

$$x \succ y \Leftrightarrow U(x) > U(y), \forall x, y \in X \quad (1)$$

Pour la fonction d'utilité globale, nous considérons le modèle transitif décomposable de Krantz, Luce, Suppes & Tversky [107] (voir aussi [108] et [109]), dans lequel U est défini par :

$$U(x) := F(u_1(x_1), \dots, u_n(x_n)), \forall x = (x_1, \dots, x_n) \in X \quad (2)$$

Où les fonctions $U_i : X_i \rightarrow \mathbb{R}$ sont appelées les fonctions d'utilité, et $F: \mathbb{R}^n \rightarrow \mathbb{R}$, non décroissantes en chacun de ses arguments, est parfois appelée la fonction d'agrégation.

Un des principaux avantages des systèmes multi-attribut est qu'il n'y a pas de notion d'incomparabilité d'un objet par rapport à un autre. En effet, cette approche permet, contrairement aux méthodes multicritère, de toujours comparer des objets entre eux. Et cela est possible grâce au score calculé pour chaque objet, reflétant son intérêt, ainsi on obtient un pré-ordre total, et ce quel que soit les objets rencontrés et leurs caractéristiques intrinsèques et extrinsèques. De plus, la théorie de l'utilité multi-attribut (MAUT) offre la possibilité d'intégrer à la fois des critères quantitatifs et qualitatifs, ainsi que d'aboutir à une hiérarchisation stricte des solutions en présence. Elle permet donc de réduire l'incertitude liée au choix d'implantation (information incomplète sur les ressources génériques et spécifiques aux différents sites) et à la rationalité du décideur (préférences subjectives).

1.6 Apprentissage automatique

Connaître les préférences d'un utilisateur ou d'un décideur est essentiel pour proposer des solutions à ses problèmes. Afin de lui faire de bonnes recommandations, un système d'aide à la décision doit continuellement apprendre ses préférences. Pour cela, on peut faire appel aux nombreuses techniques d'apprentissage automatique. Généralement, ces différentes techniques sont regroupées en trois grandes familles que nous allons présenter : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement [110].

1.6.1 Apprentissage supervisé

L'apprentissage supervisé est un ensemble de méthodes utilisant un oracle pour aider le système apprenant. L'oracle étiquette les données grâce à une fonction inconnue du système apprenant. Ces étiquettes sont un ensemble de couples entrée-sortie et constituent une base de

données d'apprentissage. Le but du système apprenant est alors de se servir de cette base de données pour se rapprocher le plus possible de la fonction de l'oracle. Il est donc possible d'identifier deux phases distinctes :

- la phase d'entraînement où le système apprenant utilise la base de données d'apprentissage de l'oracle et essaie de déterminer une fonction permettant de prédire les étiquettes données par l'oracle.
- la phase de classification où le système apprenant étiquette des données vierges en fonction de ce qu'il a appris lors de la première étape.

Ces deux phases peuvent être exécutées de façon consécutive, la phase d'entraînement puis la phase de classification. Dans ce cas, on parle d'apprentissage supervisé hors ligne. Mais les deux phases peuvent aussi être exécutées de manière itérative où le système apprenant reviendra à la phase d'entraînement après avoir commencé celle de classification. Cela permet d'affiner l'apprentissage et d'améliorer la base de données d'apprentissage avec le temps. Dans ce cas-là, on parle d'apprentissage supervisé en ligne.

L'apprentissage supervisé permet de résoudre deux types de problèmes. Les problèmes de régression où la valeur de sortie que le système apprenant cherche à estimer est une valeur dans un ensemble continu de réels et les problèmes de classification où dans ce cas, l'ensemble de valeurs de sortie est fini. Il existe un grand nombre de méthodes utilisant l'apprentissage supervisé comme les méthodes des k plus proches voisins, les réseaux de neurones ou les arbres de décision [110].

Nous allons maintenant présenter l'apprentissage non supervisé dont les techniques n'utilisent pas d'oracle pendant l'apprentissage.

1.6.2 Apprentissage non supervisé

Contrairement aux méthodes d'apprentissage supervisé, celles d'apprentissage non supervisé n'utilisent pas d'oracle pour les assister lors de l'apprentissage. Le but du système apprenant est alors de découvrir des catégories et de trouver les règles servant à catégoriser les données. L'apprentissage non supervisé ne comporte qu'une seule tâche : le regroupement de données similaires dans des groupes homogènes. La difficulté consiste à reconnaître la structure d'une donnée et la placer dans le bon groupe sans avoir dans ce cas d'étiquette sur la donnée. Contrairement à l'apprentissage supervisé, l'apprentissage non supervisé n'a pas besoin de connaissance préalable sur les sorties. Les méthodes les plus connues utilisant l'apprentissage non supervisé sont l'algorithme des k-moyennes et l'analyse en composantes.

Le problème de l'apprentissage non supervisé est qu'il fonctionne généralement en mode hors ligne. L'apprentissage est fixe et ne peut pas s'adapter au fil du temps. L'apprentissage par renforcement permet cette adaptation dans le temps.

1.6.3 Apprentissage par renforcement

L'apprentissage par renforcement regroupe des méthodes qui se servent des interactions entre un agent et son environnement pour apprendre le comportement à adopter selon la situation. A chaque action effectuée, une récompense qui peut être négative ou positive va être donnée à l'agent. Grâce à ces récompenses, l'agent va apprendre quelles actions sont les plus bénéfiques et donc quel est le comportement le plus satisfaisant. C'est un apprentissage en ligne où le système va affiner ses connaissances au cours du temps et des interactions. Barto [111] propose la définition suivante de l'apprentissage par renforcement : L'apprentissage par renforcement est une approche informatique de l'apprentissage dans laquelle un agent essaie de maximiser le montant total de la récompense qu'il reçoit en interagissant avec un environnement complexe et incertain.

1.7 Conclusion

Nous avons retracé l'historique de l'aide à la décision dont l'origine se situe dans la période d'avant deuxième guerre mondiale. Ensuite, nous avons étudié l'aide à la décision de manière générale. Puis nous nous sommes intéressés à l'aide à la décision multicritère notamment aux deux écoles qui émergent en analyse multicritère à savoir l'école européenne et américaine. L'école américaine privilégie un score d'utilité global afin de départager les actions tandis que l'école européenne favorise les méthodes de surclassement. Il a été question aussi des systèmes interactifs d'aide à la décision qui consistent à tenir compte de l'avis de l'utilisateur au fur et à mesure de l'utilisation du système. Et enfin nous avons étudié les différents types d'apprentissage à savoir l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement.

Chapitre 2 : Opérateurs d'agrégations

Les opérateurs d'agrégation jouent un rôle capital dans les systèmes d'aide à la décision. Ils permettent notamment de synthétiser les performances locales en un score global afin de pouvoir comparer les alternatives. L'un des avantages de la plupart des opérateurs d'agrégation dans le problème MCDM est qu'il considère à la fois les critères de maximisation et de minimisation simultanément.

Dans ce chapitre, nous allons voir ce qu'est un opérateur d'agrégation, ensuite nous verrons à quoi servent les opérateurs d'agrégation. Par la suite différents opérateurs d'agrégation sont discutés. Il est aussi question de la classification des opérateurs d'agrégation ainsi que de leurs propriétés générales.

2.1 Qu'est-ce qu'un opérateur d'agrégation ?

Un opérateur d'agrégation peut être considéré comme une fonction mathématique qui prend une entrée, appelée l'argument, et produit une sortie unique représentant le score global de l'entrée. Chaque entrée a une unique sortie associée à celle-ci. La fonction d'agrégation est généralement désignée par $y = f(x)$, où x est l'argument et y est la valeur agrégée. L'argument x peut être un vecteur de taille n : $x = (x_1, x_2, \dots, x_n)$, et x_1, x_2, \dots, x_n sont appelés composants de x .

Il est important de comprendre que f peut être représenté de plusieurs manières:

- a) comme une formule algébrique. Exemple : $f(x) = x_1 + x_2 - x_3$,
- b) en tant qu'une représentation graphique de la fonction (par exemple, en 2D, 3D ou en tracé de contour),
- c) verbalement, comme une séquence d'étapes (p. ex. prendre la moyenne des composantes de x), ou plus formellement, comme un algorithme,
- d) comme une table de recherche,
- e) comme solution à une équation (algébrique, différentielle ou fonctionnelle),
- f) en tant que sous-programme informatique qui renvoie une valeur y pour tout x spécifié (donc appelé oracle).

Opérateurs d'agrégations

Certaines représentations sont plus adaptées à l'analyse visuelle ou mathématique, alors que pour l'utilisation dans un programme informatique toutes les représentations (sauf graphique) sont équivalentes, car elles sont toutes finalement converties en représentation f). Certaines personnes pensent à tort que les fonctions ne sont que des formules algébriques. Nous ne ferons pas de distinction entre les fonctions selon leur représentation, f peut être donné sous la forme des différentes représentations ci-dessus. Ce qui est important est que f renvoie systématiquement la même valeur et unique pour un x donné. Les fonctions d'agrégation sont des fonctions avec des propriétés spéciales.

Le but des fonctions d'agrégation (appelées aussi opérateurs d'agrégation) est de combiner des entrées généralement interprétées comme des degrés d'appartenance à des ensembles flous, des degrés de préférence, de force probante ou de soutien d'une hypothèse, et ainsi de suite. Considérons les exemples prototypiques suivants.

Exemple 2.1 (Problème de prise de décision multicritère). Il y a deux (ou plus) alternatives, et n critères pour évaluer chaque alternative (ou plutôt une préférence pour chaque alternative). Indiquer les scores (préférences) par x_1, x_2, \dots, x_n et y_1, y_2, \dots, y_n pour les alternatives x et y respectivement. Le but est de combiner ces scores en utilisant une fonction d'agrégation f, et de comparer les valeurs $f(x_1, x_2, \dots, x_n)$ et $f(y_1, y_2, \dots, y_n)$ pour déterminer la meilleure alternative.

Exemple 2.2 (Connectives en logique floue). Un objet x a des degrés partiels d'appartenance à n ensembles flous, notés $\mu_1, \mu_2, \dots, \mu_n$. L'objectif est d'obtenir la valeur d'appartenance globale dans l'ensemble flou combiné $\mu = f(\mu_1, \mu_2, \dots, \mu_n)$. La combinaison peut être l'opération de l'union, de l'intersection, ou une opération compliquée (par exemple, composite).

Exemple 2.3 (Un problème de prise de décision de groupe). Il y a deux (ou plus) alternatives, et n décideurs ou experts qui expriment leur évaluation sur chaque alternative comme x_1, x_2, \dots, x_n . L'objectif est de combiner ces évaluations en utilisant une fonction d'agrégation f, pour obtenir un score global $f(x_1, x_2, \dots, x_n)$ pour chaque alternative.

2.2 À quoi servent les opérateurs d'agrégation?

L'objectif d'un opérateur d'agrégation est d'évaluer en général le score global d'une alternative, à partir de ses performances locales et des préférences de l'utilisateur, en vue de la comparer à d'autres alternatives dans une problématique d'aide à la décision. Ce score global permet d'établir un classement afin d'aider le décideur dans sa prise de décision.

Considérons les situations suivantes. Nous avons plusieurs critères avec lesquels nous évaluons différentes options (ou objets), et chaque option remplit chaque critère seulement partiellement (et a un score sur l'échelle $[0, 1]$). Notre objectif est d'évaluer le score combiné pour chaque option (éventuellement pour classer les options). Nous pouvons décider de faire la moyenne de tous les scores. Ceci est une approche typique de compétitions sportives (comme le patinage artistique) où les critères sont des scores donnés par des juges. Le score total est la moyenne arithmétique.

$$f(x_1, \dots, x_n) = \frac{1}{n}(x_1 + x_2 + \dots + x_n) \quad (3)$$

Nous pouvons décider d'adopter une approche différente: les scores faibles tirent le score global vers le bas. Le score global ne sera pas supérieur au minimum des scores individuels des critères. Ceci est un exemple de comportement conjonctif. Les fonctions d'agrégation conjonctive sont appropriées pour modéliser des conjonctions comme

$$\text{I f } x \text{ i s } A \text{ A N D } y \text{ i s } B \text{ A N D } z \text{ i s } C \text{ T H E N } \dots \quad (4)$$

Où A, B et C sont les critères selon lesquels les paramètres x, y, z sont évalués. Par exemple pour choisir les astronautes, les candidats doivent remplir différents critères et la note globale est déterminée en prenant la plus mauvaise note parmi les critères d'évaluation. Les candidats doivent donc avoir de bonnes notes partout afin d'espérer être sélectionné.

Plusieurs opérateurs d'agrégation existent et nous en expliciterons quelques-uns dans cette partie.

2.3 Quelques opérateurs d'agrégation

Il existe de nombreux opérateurs d'agrégation, nous ne présenterons dans cette section qu'une sélection de ces opérateurs.

2.3.1 La somme pondérée

La somme pondérée est un opérateur d'agrégation souvent utilisé pour sa simplicité de mise en œuvre. Elle demande d'attribuer un poids à chaque critère en fonction de son importance dans un problème de décision. Elle est définie par :

$$\psi(a_1, \dots, a_n) = \sum_{i=1}^n w_i a_i \quad (5)$$

avec $w_i \in [0, 1]$ les poids des critères tels que

$$\sum_{i=1}^n w_i = 1 \quad (6)$$

La somme pondérée est un opérateur très limité du fait qu'il ne permet pas de prendre en compte les dépendances entre les critères. Par ailleurs, les préférences du décideur sont prises en compte de manière simpliste, à travers de simples poids affectés à chaque critère.

Exemple : Choix d'une voiture du père de famille [95]

Tableau 1 : choix de la voiture du père de famille, évaluation des alternatives

Modèle de voiture	Confort	Sécurité	Indice de coût kilométrique	Esthétique	Vitesse de pointe	Prix d'achat
A1	Dans l'ensemble satisfaisant mais place légèrement insuffisante	Normale	0.39	Convenable	140km/h	0.87*B
A2	Particulièrement satisfaisant et spacieux	Normale	0.41	Elégant	145km/h	0.95*B
A3	Acceptable mais étroit	Normale	0.66	Très élégant	185km/h	0.99*B
A4	Particulièrement satisfaisant et spacieux	Supérieur à la normale	0.40	Elégant	145km/h	Vraisemblablement > B

Après transformation en utilisant l'approche classique de notation d'une échelle de 1 à 10 max nous obtenons le tableau suivant :

Tableau 2 : Choix de la voiture du père de famille, matrice de performance

Modèle de voiture	Confort	Sécurité	Indice de coût kilométrique	Esthétique	Vitesse de pointe	Prix d'achat
A1	6	5	7	5	6	9
A2	8	5	6	7	7	8
A3	5	5	5	9	9	7
A4	8	7	6	7	7	2

Tableau 3 : choix de la voiture du père de famille, poids des critères

Confort	Sécurité	Indice de coût kilométrique	Esthétique	Vitesse de pointe	Prix d'achat
4	3	4	2	3	5

Après agrégation en utilisant la somme pondérée, nous obtenons le résultat suivant :

Tableau 4 : choix de la voiture du père de famille, résultats

Modèle de voiture	Score
A1	$(6*4+5*3+7*4+5*2+6*3+9*5)/21$ 6.67
A2	$(8*4+5*3+6*4+7*2+7*3+8*5)/21$ 6.95
A3	$(5*4+5*3+5*4+9*2+9*3+7*5)/21$ 6.43
A4	$(8*4+7*3+6*4+7*2+7*3+2*5)/21$ 5.81

La meilleure alternative en appliquant la somme pondérée est l'alternative A2 avec un score global de 6.95. Ce modèle de voiture présente l'avantage d'être assez confortable et un prix assez abordable. Il dispose d'un score acceptable également pour les autres critères.

Une autre façon de faire est d'affecter des poids aux critères selon leur degré de satisfaction et non selon leur nature. C'est ce que fait la somme pondérée ordonnée que nous allons voir par la suite.

2.3.2 La somme pondérée ordonnée

La somme pondérée ordonnée [112] pour (Ordered Weighted Averaging - OWA) est une classe d'opérateurs d'agrégation qui détermine le poids du critère en fonction de son degré de satisfaction à la différence de la somme pondérée où le poids dépend de la nature du critère. Elle est définie par :

$$OWA_w(a_1, \dots, a_n) = \sum_{i=1}^n w_i a_{(i)} \quad (7)$$

avec $W = (w_1, \dots, w_n)$ un vecteur de poids, $w_i \in [0, 1]$ tel que

$$\sum_{i=1}^n w_i = 1 \quad (8)$$

où la notation (i) est une permutation des indices telle que $a_{(1)} \leq \dots \leq a_{(n)}$.

Il est possible de modéliser un grand nombre de fonctions de base telles que:

- la fonction « Max » avec un vecteur $W = (0, 0, \dots, 1)$
- la fonction « Min » avec un vecteur $W = (1, \dots, 0, 0)$
- la fonction « moyenne » avec pour n critères un vecteur $W = (\frac{1}{n}, \dots, \frac{1}{n})$

Exemple : reprenons l'exemple précédent à savoir le choix de la voiture du chef de famille avec l'énoncé suivant : Le poids des critères varie du moins satisfaisant (1) au plus satisfaisant (6). Aucune préférence n'est accordée à la nature des critères, il suffit d'avoir plus de bons scores pour avoir un meilleur score global.

Tableau 5 : Choix de la voiture du père de famille, matrice de performance

Modèle de voiture	Confort	Sécurité	Indice de coût kilométrique	de Esthétique	Vitesse de	Prix d'achat
A1	6	5	7	5	6	9
A2	8	5	6	7	7	8
A3	5	5	5	9	9	7
A4	8	7	6	7	7	2

Tableau 6 : Matrice de performance ordonnée et résultats

Modèle de voiture							Score global
A1	5	5	6	6	7	9	6.95
A2	5	6	7	7	8	8	7.33
A3	5	5	5	7	9	9	7.47
A4	2	6	7	7	7	8	6.95
Poids	1	2	3	4	5	6	

Nous constatons qu'en utilisant la somme pondérée ordonnée l'alternative A3 obtient le meilleur score. Car elle a deux excellentes notes qui ont des poids 5 et 6.

Opérateurs d'agrégations

Ces deux opérateurs d'agrégation que sont la somme pondérée et la somme pondérée ordonnée, considèrent les critères indépendants entre eux, ce qui est malheureusement peu fréquent dans la réalité. En effet, dans la vie réelle, les critères sont très souvent dépendants les uns et des autres. La prise en compte de ces interactions entre critères est fondamentale pour une prise de décision robuste et reflétant la réalité. D'autres opérateurs d'agrégation tels que l'intégrale de Choquet, permettent de prendre en compte ces interactions entre les critères que nous allons décrire par la suite.

2.3.3 L'intégrale de Choquet

Les opérateurs d'agrégation tels que la somme pondérée ou OWA ne sont pas capables de modéliser les interactions entre les critères. Hors, il est très rare qu'il n'y ait pas de synergie entre les critères. Afin d'obtenir une représentation plus proche de la réalité, ces interactions doivent être prises en compte et pour cela, l'utilisation d'un vecteur poids n'est pas appropriée. Remplacer ce vecteur poids par une fonction non-additive permet en plus de définir un poids pour chaque critère, de définir aussi un poids pour chaque sous-ensemble de critères. L'utilisation de ces fonctions non-additives permet d'avantage d'exprimer la subjectivité humaine et de modéliser les synergies positives ou négatives entre les critères. Sugeno a utilisé ces fonctions non-additives et a proposé de les appeler mesures floues [113].

Définition. Une mesure floue sur N est une fonction d'ensemble $\mu : 2^N \rightarrow [0, 1]$ qui est monotone, c'est à dire $\mu(S) \leq \mu(T)$ chaque fois que $S \subseteq T$, et vérifie les conditions limites $\mu(\emptyset) = 0$ et $\mu(N) = 1$

$\mu(S)$ est considéré comme étant le poids ou l'importance du sous-ensemble de critère S . Compte tenu de la monotonie des fonctions floues, si on ajoute un élément i à l'ensemble S , le poids de l'ensemble $S + i$ ne pourra être plus faible que le poids de S . Utiliser les mesures floues nécessite de déterminer 2^N poids correspondants aux 2^N sous-ensembles de N . Dans le cas où une mesure floue est additive, c'est à dire si $\mu(S \cup T) = \mu(S) + \mu(T)$, seuls les poids des n critères sont suffisants pour calculer la mesure floue.

Les mesures floues peuvent être utilisées comme opérateurs d'agrégation. En utilisant les poids des critères et des ensembles de critères, il est alors possible de représenter les interactions entre les critères. Ces fonctions sont appelées des intégrales floues [113]. Il existe

Opérateurs d'agrégations

plusieurs classes d'intégrales floues dont l'une des plus représentatives est l'intégrale de Choquet [114].

L'intégrale de Choquet est définie par :

Soit μ une mesure floue sur N . L'intégrale de Choquet de $x \in R^n$ par rapport à μ est définie par :

$$C_\mu(x) = \sum_{i=1}^n x(i) [\mu(A(i)) - \mu(A(i+1))] \quad (9)$$

où $(.)$ indique une permutation sur N telle que $x_{(1)} \leq \dots \leq x_{(n)}$. D'autre part, $A(i) = \{(i), \dots, (n)\}$ et $A(n+1) = \emptyset$

L'utilisation des mesures floues dans l'intégrale de Choquet permet de comprendre les dépendances entre les critères mais aussi l'importance que peut avoir chaque critère

L'indice d'importance [115] ou valeur de Shapley du critère i par rapport à μ est défini par :

$$\Phi(\mu, i) := \sum_{T \subseteq N \setminus i} \frac{(n-t-1)! t!}{n!} [\mu(T \cup i) - \mu(T)] \quad (10)$$

Dans le cas où μ est additif, nous avons $\mu(T \cup i) - \mu(T) = \mu(i)$. Dans le cas contraire, cette égalité n'est plus vraie et les critères sont alors dépendants. Il est alors intéressant de connaître le degré d'interaction entre deux critères. Cet indice d'interaction [116] entre les critères i et j par rapport à μ est défini par :

$$I(\mu, ij) = \sum_{T \subseteq N \setminus ij} \frac{(n-t-2)! t!}{(n-1)!} (\Delta_{ij}\mu)(T) \quad (11)$$

$$\text{avec } (\Delta_{ij}\mu)(T) := \mu(T \cup ij) - \mu(T \cup i) - \mu(T \cup j) + \mu(T). \quad (12)$$

L'indice d'interaction $I(\mu, ij)$ est compris dans l'intervalle $[-1, 1]$ pour tout $i, j \in N$. Si cet indice est positif, alors il y a une synergie entre ces deux critères. A l'inverse si l'indice d'interaction est négatif, on parle alors de critères redondants.

Examinons avec plus d'attention le modèle 2-additif. Celui-ci est particulièrement intéressant car il permet de représenter des interactions entre 2 critères. De plus, il n'est pas nécessaire de calculer toutes les paramètres de la capacité entre plusieurs critères. On montre que dans ce cas, l'intégrale de Choquet se met sous la forme suivante, pour $a \in R^n$:

$$C_\mu(a) = \sum_{I_{ij} > 0} (a_i \wedge a_j) I_{ij} + \sum_{I_{ij} < 0} (a_i \vee a_j) |I_{ij}| + \sum_{i=1}^n a_i (\Phi(i) - \frac{1}{2} \sum_{j \neq i} |I_{ij}|) \quad (13)$$

On remarque que si $I_{ij} > 0$, alors l'agrégation est du type *conjonctif* (critères reliés par un minimum), et que si $I_{ij} < 0$, alors l'agrégation est de type *disjonctif* (critères reliés par un maximum). Enfin, il reste un terme linéaire faisant intervenir la valeur de Shapley. On peut montrer que $C_\mu(a)$ est une combinaison convexe des a_i (dictateurs), des termes $a_i \wedge a_j$ (conjonctions) et $a_i \vee a_j$ (disjonctions) [117].

2.3.4 L'intégrale de Sugeno

Contrairement à l'intégrale de Choquet qui utilise des évaluations quantitatives, l'intégrale de Sugeno est utilisée pour des évaluations qualitatives. L'intégrale de Sugeno a été introduite dans [113]. Nous considérons ici l'intégrale de Sugeno dans sa version discrète, appliquée à l'agrégation de préférence. Nous considérons également un ensemble L totalement ordonné, pas nécessairement un ensemble numérique, appelé échelle d'évaluation, et dont les éléments minimum et maximum sont notés respectivement 0 et 1. L'intégrale de Sugeno est définie par rapport à une capacité de l'ensemble N qui est une fonction:

$$\mu : 2^N \rightarrow L \text{ tel que : } \mu(\emptyset) = 0 \text{ et } \mu(N) = 1, \text{ pour } I \subseteq J \subseteq N : \mu(I) \leq \mu(J).$$

Pour tout ensemble de critères $I \subseteq N$, la valeur de $\mu(I)$ peut être interprétée comme le degré d'importance associé à I . Soit une capacité $\mu : 2^N \rightarrow L$. L'intégrale de Sugeno définie par rapport à μ , notée S_μ , est exprimé sous la forme :

$$S_\mu(y_1, y_2, \dots, y_n) = \bigvee_{i=1}^n \left(y_{(i)} \bigwedge \mu(\{(i), \dots, (n)\}) \right) \quad (14)$$

où (i) désigne la permutation des composantes de $y = (y_1, \dots, y_n)$ telle que $y_{(1)} \leq \dots \leq y_{(n)}$. Par cette formule, on voit en particulier que μ détermine S_μ entièrement et uniquement. Le problème d'apprentissage d'une intégrale de Sugeno S_μ peut donc être réduit à celui d'apprentissage de la capacité correspondante μ .

Nous allons voir une série d'opérateurs d'agrégation dont les évaluations sont quantitatives et sont proches de la somme pondérée.

2.3.5 Multi-objective optimization based on ratio analysis (MOORA)

MOORA, développé par Brauers et Zavadskas [118], est une méthode MCDM comprenant deux phases, à savoir l'approche du point de référence et l'approche du système de ratio, et permet de mesurer des critères bénéfiques et non bénéfiques dans un processus de sélection avec un ensemble d'alternatives. Une matrice de décision initiale dont le kj -ième élément affiche l'évaluation de la performance de la k -ième alternative ($k = 1, \dots, t$) sur le j -ième critère de décision ($j = 1, \dots, n$) est formée. Par conséquent:

Étape 1. Normalisation de la matrice de décision. Pour obtenir des éléments sans dimension et comparables dans le processus d'évaluation, les éléments kj -ième de la matrice initiale sont normalisés en utilisant les équations suivantes:

$$r_{kj} = \frac{x_{kj}}{\sqrt{\sum_{k=1}^t x_{kj}^2}} \quad (15)$$

Étape 2. Détermination de la matrice normalisée pondérée. L'élément kj -ième de la matrice normalisée est remplacé par celui calculé en utilisant ce qui suit:

$$v_{kj} = r_{kj} \cdot w_j \quad (16)$$

Étape 3. Calculer la note globale des critères positif et négatif pour chaque alternative. La note globale de la k -ième alternative considérant les critères bénéfiques et non-bénéfiques est calculée en appliquant les équations. (17) et (18), respectivement

$$S_k^+ = \sum_{j \in J^{Max}} v_{kj} \quad (17)$$

Où J^{Max} est l'ensemble d'indices de l'ensemble des critères bénéfiques pour lesquels des valeurs plus élevées sont souhaitables;

$$S_k^- = \sum_{j \in J^{Min}} v_{kj} \quad (18)$$

Où J^{Min} est l'ensemble d'indices de l'ensemble des critères non bénéfiques pour lesquels les valeurs inférieures sont préférables.

Étape 4. Évaluer la performance globale de chaque alternative. La performance globale de la k ième alternative est calculée comme la différence entre les notes globales pour les critères bénéfiques et non bénéfiques:

$$S_k = S_k^+ - S_k^- \quad (19)$$

Étape 5. Classement des alternatives. Les valeurs forment une échelle cardinale qui peut être utilisée pour comparer et classer les alternatives: plus la valeur de l'alternative est élevée, plus elle est préférée dans le classement final.

2.3.6 Complex proportional assessment of alternatives (COPRAS)

COPRAS est une autre méthode MCDM qui sélectionne la meilleure alternative parmi beaucoup d'alternatives réalisables en déterminant une solution avec rapport direct et proportionnel à la meilleure solution au rapport avec la solution idéale-pire. La méthode COPRAS a été appliquée dans le système d'information géographique, l'évaluation du cycle de vie du bâtiment et le problème de sélection des matériaux (Zavadskas, 2007, Banaitiene, 2008, Chatterjee, 2011). Pour résoudre un problème MCDM par COPRAS, après avoir déterminé les alternatives et les critères associés, suivez les étapes ci-dessous;

Étape 1 - Normaliser la matrice de décision: supposons que la matrice de décision de l'alternative soit sous le critère d'évaluation, puis la matrice de décision normalisée est ici;

$$r_{ij} = \frac{x_{ij}}{\sum_{j=1}^m x_{ij}}, \quad j = 1, 2, \dots, m, \quad i = 1, 2, \dots, n \quad (20)$$

Étape 2 – Calculer la matrice de décision normalisée pondérée où figure le poids des critères et donnée par $\sum_{i=1}^n w_i = 1$;

$$v_{ij} = w_i \times r_{ij}, \quad j = 1, 2, \dots, m, \quad i = 1, 2, \dots, n \quad (21)$$

Étape 3-Calculer la matrice de décision normalisée pondérée (P_j) où figure le poids des critères et donnée par

$$P_j = \sum_{i=1}^k v_{ij} \quad (22)$$

Où k est le nombre de critères de valeur, qui doit être maximisé

Étape 4 - Obtenir les sommes des valeurs de critères normalisées pondérées (R_j) pour chaque variante dont les valeurs plus petites sont préférables en utilisant l'équation suivante;

$$R_j = \sum_{i=1}^{n-k} v_{ij} \quad (23)$$

Où $(n-k)$ est le nombre de valeurs de critères, qui devrait être minimisé

Étape 5 - Identifier le poids relatif de chaque alternative Q_j

$$Q_j = P_j + \frac{\sum_{j=1}^m R_j}{R_j \sum_{j=1}^m \frac{1}{R_j}}, \quad (24)$$

Étape 7 - Déterminer la priorité des alternatives en fonction des valeurs de Q_j . Plus la valeur de Q_j est élevée, plus l'alternative est mieux classé dans le classement final.

2.3.7 Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)

TOPSIS est une méthode dont le but est de pouvoir classer par ordre de choix un certain nombre d'alternatives sur la base d'un ensemble de critères favorables ou défavorables. Elle a été développée par Hwang et Yoon en 1981. Son principe consiste à déterminer pour chaque alternative un coefficient compris entre 0 et 1 sur la base des distances euclidiennes entre chaque alternative d'une part et les solutions idéales favorables et défavorables d'autre part

Opérateurs d'agrégations

Étape 1: Normaliser la matrice de décision: x_{ij} est la matrice de décision de l'alternative j sous le critère d'évaluation i , puis la matrice de décision normalisée est :

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{j=1}^m x_{ij}^2}}, \quad j = 1, 2, \dots, m, \quad i = 1, 2, \dots, n \quad (25)$$

Étape 2 - Construire la matrice de décision normalisée pondérée où w_i sont les poids des critères et donnés par

$$\sum_{i=1}^n w_i = 1$$

$$v_{ij} = w_i \times r_{ij}, \quad j = 1, 2, \dots, m, \quad i = 1, 2, \dots, n \quad (26)$$

Étape 3- Déterminer la solution idéale positive et négative (PIS) et (NIS) pour la matrice de décision normalisée pondérée. Ces valeurs sont désignées par les relations suivantes;

$$A^* = \{v_1^*, v_2^*, \dots, v_i^*\} = \{(Max v_{ij} | i \in I'), (Min v_{ij} | i \in I'')\} \quad (27)$$

$$A^- = \{v_1^-, v_2^-, \dots, v_i^-\} = \{(Min v_{ij} | i \in I'), (Max v_{ij} | i \in I'')\} \quad (28)$$

Où I' est associé aux critères bénéfiques, et I'' est associé aux critères non bénéfiques.

Étape 4- Calculez les mesures de séparation. La séparation de chaque alternative de la solution idéale positive (D_j^*) et de la solution idéale négative peut être mesurée en utilisant la distance euclidienne n -dimensionnelle comme ici;

$$D_j^* = \sqrt{\sum_{i=1}^n (v_{ij} - v_i^*)^2}, \quad j = 1, 2, \dots, m, \quad D_j^- = \sqrt{\sum_{i=1}^n (v_{ij} - v_i^-)^2}, \quad j = 1, 2, \dots, m \quad (29)$$

Étape 5 - Calcul de la proximité relative à la solution idéale où la valeur d'indice CC_j^* est comprise entre 0 et 1.

$$CC_j^* = \frac{D_j^-}{D_j^* + D_j^-}, \quad j = 1, 2, \dots, m \quad (30)$$

Étape 6 - Classement de la priorité en fonction des valeurs décroissantes CC^*_j La plus grande valeur de l'indice signifie une meilleure performance des alternatives.

2.3.8 Weighted aggregated sum product assessment (WASPAS)

WASPAS est l'une des méthodes MCDM les plus récemment développées. C'est une combinaison de deux approches MCDM bien connues, à savoir le modèle de somme pondérée (WSM) et le modèle de produit pondéré (WPM). La matrice suivante est utilisée pour démarrer la procédure de solution:

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{t1} & x_{t2} & \dots & x_{tn} \end{bmatrix} \quad (31)$$

où t est le nombre d'alternatives, n est le nombre de critères d'évaluation et, pour $k = 1, \dots, t$ et $j = 1, \dots, n$, x_{kj} est l'indice de performance de la k -ième alternative sur le j -ième critère. Cette matrice de décision est normalisée en utilisant les équations suivantes où l'élément générique normalisé de la matrice de décision est désigné par \tilde{x}_{kj} .

Pour les critères bénéfiques:

$$\tilde{x}_{kj} = \frac{x_{kj}}{\max_k x_{kj}} \quad (32)$$

Pour les critères non bénéfiques

$$\tilde{x}_{kj} = \frac{\min_k x_{kj}}{x_{kj}} \quad (33)$$

L'algorithme WASPAS est un algorithme de recherche optimal basé sur deux critères d'optimisation. Le premier critère est un critère de succès moyen pondéré similaire à la méthode WSM et le second critère est basé sur la méthode WPM. Sur la base des méthodes WSM et WPM, l'importance relative totale de la k -ième alternative est calculée comme suit

$$Q_k^{(1)} = \sum_{j=1}^n \tilde{x}_{kj} w_j \quad (34)$$

$$Q_k^{(2)} = \prod_{j=1}^n (\tilde{x}_{kj})^{w_j} \quad (35)$$

Un critère général d'agrégation pondérée des méthodes additives et multiplicatives peut alors être proposé comme suit:

$$Q_k = 0.5Q_k^{(1)} + 0.5Q_k^{(2)} = 0.5 \sum_{j=1}^n \tilde{x}_{kj} w_j + 0.5 \prod_{j=1}^n (\tilde{x}_{kj})^{w_j} \quad (36)$$

Afin d'augmenter la précision du classement et l'efficacité du processus de prise de décision, une équation plus généralisée pour déterminer l'importance relative totale de la k-ième alternative peut être utilisée:

$$Q_k = \lambda Q_k^{(1)} + (1-\lambda)Q_k^{(2)} = \lambda \sum_{j=1}^n \tilde{x}_{kj} w_j + (1-\lambda) \prod_{j=1}^n (\tilde{x}_{kj})^{w_j} \quad (37)$$

Enfin, les alternatives candidates peuvent être classées en fonction des valeurs de Q , c'est-à-dire que la meilleure alternative serait celle ayant la valeur la plus élevée. Lorsque la valeur de λ est 0, la méthode WASPAS coïncide avec WPM, tandis que pour $\lambda=1$, WASPAS correspond à WSM.

2.3.9 Evaluation based on Distance from Average Solution (EDAS)

La meilleure alternative dans la méthode EDAS correspond à la distance de la solution moyenne (AV). Dans cette méthode, deux premières mesures sont fournies: la distance positive par rapport à la moyenne (PDA) et la distance négative par rapport à la moyenne (NDA). Ces mesures peuvent montrer la différence entre chaque solution (alternative) et la solution moyenne. Par conséquent, des valeurs plus élevées de PDA et des valeurs inférieures de NDA indiqueront une solution optimale. Les valeurs plus élevées de PDA et / ou les valeurs inférieures de NDA représentent que la solution (alternative) est meilleure que la solution moyenne. L'algorithme classique d'EDAS peut être suivi en suivant les étapes suivantes.

Étape 1 - Choisissez les critères les plus pertinents qui décrivent les alternatives de décision pour un problème de décision spécifique

Étape 2 – Si x_{ij} est la performance de la i-ième alternative A_1, A_2, \dots, A_n , ($i=1,2,\dots, n$) en respectant le j-ième critère C_1, C_2, \dots, C_m , ($j=1,2,\dots, m$) Ainsi, pour former la matrice de

Opérateurs d'agrégations

décision d'intervalle et le poids de chaque critère, le tableau suivant et les variables doivent être considérés;

$$X = [x_{ij}]_{n \times m} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix}, W = [w_1, w_2, \dots, w_m] \quad (38)$$

Pour $(i = 1, 2, \dots, n)$ et $(j = 1, 2, \dots, m)$

où w_j est le poids j^{th} critère.

Étape 3- Selon la définition de la méthode EDAS, la solution moyenne par rapport à tous les critères doit être déterminée comme indiqué dans les formules;

$$AV_j = \frac{\sum_{i=1}^n x_{ij}}{n} \quad (39)$$

Étape 4 - La distance positive par rapport à la moyenne (PDA) et la distance négative par rapport aux matrices moyennes (NDA) doivent être calculées dans cette étape en fonction des valeurs inférieures et supérieures de la matrice, comme indiqué:

$$PDA_{ij} = \frac{\max(0, (x_{ij} - AV_j))}{AV_j} \quad (40)$$

$$NDA_{ij} = \frac{\max(0, (AV_j - x_{ij}))}{AV_j} \quad (41)$$

De cette manière PDA_{ij} et NDA_{ij} représentent la distance positive et négative de la i -ième alternative moyenne en termes de critère pour le niveau inférieur de la matrice de décision, respectivement.

Étape 5 - Obtenir la somme pondérée de la distance positive et des distances négatives par rapport au matriciel moyen:

$$SP_i = \sum_{j=1}^m w_j PDA_{ij} \quad (42)$$

$$SN_i = \sum_{j=1}^m w_j NDA_{ij} \quad (43)$$

Étape 6 - Identifier les valeurs normalisées de SP_i et SN_i pour toutes les alternatives, représentées comme suit:

$$NSP_i = \frac{SP_i}{Max_i(SP_i)} \quad (44)$$

$$NSN_i = 1 - \frac{SN_i}{Max_i(SN_i)} \quad (45)$$

Étape 7 - Détecter le score d'évaluation pour toutes les alternatives, montré comme suit:

$$AS = \frac{1}{2}(NSP_i + NSN_i), \quad (46)$$

où $0 \leq AS \leq 1$

Étape 8 - Classez les alternatives en fonction des valeurs décroissantes du score d'évaluation (AS). L'alternative avec le plus haut score est le meilleur choix parmi les alternatives candidates.

2.4 Classification des opérateurs d'agrégation

Il existe différentes sémantiques d'agrégation, et les classes principales sont déterminées en fonction de ces sémantiques. Dans certains cas, nous exigeons que les entrées se comportent comme des moyennes et dans d'autres cas, les fonctions d'agrégation modélisent les connecteurs logiques (disjonction et conjonction), de sorte que les entrées se renforcent mutuellement, et parfois le comportement des fonctions d'agrégation dépend des entrées. Les quatre classes principales de fonctions d'agrégation sont :

- Compromis,
- Conjonctif,
- Disjonctif,
- Mixte.

2.4.1 Opérateurs de compromis :

Les opérateurs de compromis, mènent à un score qui sera compris entre le minimum et le maximum. Le but ici est d'obtenir une sorte de compromis entre tous les arguments en entrée, l'influence d'un faible score pouvant être atténué par un score plus élevé et inversement.

Une fonction d'agrégation f a un comportement de moyenne (ou fait la moyenne) si pour chaque x elle est bornée par

$$\min(x) \leq f(x) \leq \max(x) \quad (47)$$

Clairement, les opérateurs de compromis ne sont ni conjonctifs ni disjonctifs. L'exemple le plus connu et sûrement le plus utilisé d'un tel opérateur est certainement la moyenne arithmétique. Parmi ces opérateurs, on trouvera aussi les opérateurs de type moyenne et les statistiques d'ordre. D'autres opérateurs rentrant dans cette catégorie seront également décrits par la suite (opérateurs OWA, intégrales floues). On pourra par ailleurs trouver une étude détaillée du comportement optimiste ou pessimiste de ces opérateurs dans le cadre de la prise de décision dans [119].

2.4.2 Agrégation conjonctive.

Les opérateurs conjonctifs modélisent une agrégation connectant chacun des arguments par une opération correspondant au et logique. En d'autres termes, le score final ne sera élevé que si l'ensemble des arguments est élevé, ce qui correspondra finalement à un comportement intolérant, puisque tous les critères doivent être satisfaits pour engendrer un score positif.

Définition : Une fonction d'agrégation f a un comportement conjonctif (ou est conjonctive) si pour chaque x elle est délimitée par :

$$f(x) \leq \min(x) = \min(x_1, x_2, \dots, x_n) \quad (48)$$

Opérateurs d'agrégations

Si à ce comportement on ajoute des propriétés mathématiques de non décroissance (monotonie), commutativité (symétrie) et associativité que l'on a énoncée en amont, on obtient alors une famille bien connue d'opérateurs d'agrégation : les normes triangulaires

2.4.3 Agrégation disjonctive.

Les opérateurs disjonctifs modélisent une agrégation connectant chacun des arguments par une opération correspondant au ou logique. En d'autres termes, le score final sera haut si au moins un des arguments est haut, ce qui correspondra finalement à un comportement tolérant, puisque un seul critère satisfait suffit pour engendrer un score positif.

Définition : une fonction d'agrégation f a un comportement disjonctif (ou est disjonctive) si pour chaque x elle est délimitée par :

$$f(x) \geq \max(x) = \max(x_1, x_2, \dots, x_n) \quad (49)$$

Si à ce comportement on ajoute des propriétés mathématiques de non décroissance (monotonie), commutativité (symétrie) et associativité que l'on a énoncées en amont, on obtient alors une famille bien connue d'opérateurs d'agrégation : les conormes triangulaires

2.4.4 Agrégation mixte.

Une fonction d'agrégation f est dite mixte si elle n'appartient à aucune des classes ci-dessus, c'est-à-dire qu'elle présente différents types de comportement sur différentes parties du domaine.

Tableau 7 : Classification des opérateurs évoqués

Opérateur d'agrégation	Catégorie
Somme pondérée	Compromis
Somme pondérée ordonnée	Compromis
Intégrale de Choquet	Compromis
Intégrale de Sugeno	Mixte
MOORA	Disjonctive
COPRAS	Disjonctive
TOPSIS	Compromis
WASPAS	Compromis
EDAS	Compromis

2.5 Propriétés générales

2.5.1 Idempotence

En algèbre, l'idempotence est une propriété liée à une opération $*$ par laquelle un élément x est idempotent, c'est à dire $x * x = x$. Dans le cadre des opérateurs n -aires, cette notion s'étend à la propriété suivante.

Définition . Un opérateur d'agrégation A possède un élément idempotent x si

$$A(x, \dots, x) = x \quad (50)$$

On dira qu'un opérateur d'agrégation est idempotent si la propriété (50) est respectée pour tout $x \in I=[0,1]$, l'intervalle de notation.

Cette propriété, aussi appelée unanimité, s'interprète de la manière suivante : si on agrège n fois la même valeur, alors on s'attend à obtenir cette même valeur initiale. On voit aisément que si les conditions aux bornes sont respectées, alors 0 et 1 sont des éléments idempotents, que l'on appellera triviaux. Désirer l'idempotence d'un opérateur ainsi que sa

Opérateurs d'agrégations

monotonie revient à vouloir un comportement de compensation, et l'on se restreindra dans ce cas aux opérateurs respectant $\min \leq A \leq \max$.

2.5.2 Continuité

Un opérateur d'agrégation est dit continu si la fonction d'agrégation est continue dans le sens usuel de ce terme, c'est à dire que pour tout $n \in \mathbb{N}$, l'opération n-aire A est continue. Cette propriété est souvent nécessaire dans de nombreuses applications, puisque elle contraint l'opérateur à ne pas se comporter de manière chaotique. En effet, la continuité assure qu'un changement, ou une erreur faible en entrée ne mènera pas à un grand changement, ou erreur, à la sortie.

2.5.3 Symétrie

La symétrie, ou la commutativité, d'opérateur binaires, c'est à dire la propriété $x*y=y*x$ peut facilement être étendue au cas d'opérations n-aires. Lorsque $n > 2$, on parle alors de symétrie.

Définition : On dit qu'un opérateur d'agrégation A est symétrique si

$$A(x_1, \dots, x_n) = A(x_{\sigma(1)}, \dots, x_{\sigma(n)}) \quad (51)$$

pour toute permutation σ de l'ensemble $N = \{1, \dots, n\}$.

Les opérations maximum, minimum, ou encore les moyennes sont symétriques, mais les moyennes pondérées ne le sont pas. Définir un opérateur d'agrégation symétrique revient à associer à chacune des valeurs d'entrée la même importance, ce qui pousse certains auteurs à qualifier cette propriété d'anonymat.

2.5.4 Associativité

L'associativité d'une opération binaire $*$ définie sur un domaine I signifie que le couple $(I, *)$ est un demi-groupe.

Définition : . Un opérateur d'agrégation A est associatif si

$$\begin{aligned} \forall n, n' \in \mathbb{N}, \forall x_1, \dots, x_n \text{ et } y_1, \dots, y_{n'} \in I \\ A(x_1, \dots, x_n, y_1, \dots, y_{n'}) = A(A(x_1, \dots, x_n), A(y_1, \dots, y_{n'})) \end{aligned} \quad (52)$$

Cette propriété est intéressante dans la mesure où l'on peut commencer à agréger les valeurs sans connaître l'ensemble des arguments d'entrée, ce qui peut être une contrainte d'un système en ligne par exemple. Pratiquement, on divise ainsi les entrées en tuples que l'on agrège, et l'ordre d'apparition de ces tuples lors de l'opération d'agrégation ne doit pas changer le résultat final. À partir d'un opérateur associatif défini pour deux opérands, il est donc aisé de l'étendre à son équivalent n-aire. Un exemple d'opérateur associatif est le produit. Inversement, la moyenne arithmétique n'est pas associative. On pourra se référer à [120] pour une étude détaillée des fonctions associatives.

2.5.5 Élément neutre et élément absorbant

Venant encore une fois d'une notion connue dans le cadre des opérations binaires, un élément e est appelé élément neutre de l'opération * si pour tout x, on a $x * e = e * x = x$. À partir de cette définition, on pourra donc dire que l'action d'un élément neutre a le même effet que s'il n'était pas présent.

Définition : . Un opérateur d'agrégation A possède un élément neutre e si

$$\begin{aligned} \forall n \in \mathbb{N}, \forall x_1, \dots, x_n \in I, \exists i \text{ tel que } x_i = e, \text{ alors} \\ A(x_1, \dots, x_n) = A(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) \end{aligned} \quad (53)$$

Inversement, un élément a est appelé élément absorbant de l'opération * si pour tout x, on a $x * a = a * x = a$. On transpose alors cette définition à l'opérateur d'agrégation n-aire de la façon suivante.

Définition : Un opérateur d'agrégation A possède un élément absorbant a si

$$\forall n \in \mathbb{N}, \forall x_1, \dots, x_n \in I, \exists i \text{ tel que } x_i = a, \text{ alors} \\ A(x_1, \dots, x_n) = a \quad (54)$$

C'est l'attitude contraire à l'élément neutre, dans la mesure où la simple présence de cet élément parmi les valeurs d'entrée suffit à provoquer une sortie respectant celui-ci. Cela implique donc ce comportement, ou vote, que l'on qualifiera de veto.

2.6 Propriétés des opérateurs évoqués

Tableau 8 : Propriétés des opérateurs évoqués

	Idempotence	Continuité	Symétrie	Associativité	Élément neutre	Élément absorbant
Somme pondérée	Oui	Oui	Non	Non	Non	Non
Somme pondérée ordonnée	Oui	Oui	Oui	Non	Non	Non
Intégrale de Choquet	Oui	Oui	Oui	Non	Non	Non
Intégrale de Sugeno	Non	Oui	Oui	Non	Non	Non
MOORA	Non	Oui	Non	Non	Non	Non
COPRAS	Non	Oui	Non	Non	Non	Non
TOPSIS	Non	Oui	Non	Non	Non	Non
WASPAS	Non	Oui	Non	Non	Non	Non
EDAS	Non	Oui	Non	Non	Non	Non

2.7 Conclusion

Les opérateurs d'agrégation jouent un rôle capital dans les systèmes de recommandation. Une présentation sommaire des opérateurs d'agrégation tels que la somme pondérée, la somme pondérée ordonnée, les intégrales de Choquet et Sugeno, les méthodes MOORA, COPRAS, TOPSIS etc a été faite dans cette partie. Il a été question aussi des différentes classes d'opérateur ainsi que de leurs propriétés.

Chapitre 3: Systemes de recommandation

Les systemes de recommandation sont une forme specifique de filtrage de l'information visant a presenter les elements d'information (films, musique, livres, news, images, pages Web, etc) qui sont susceptibles d'interesser l'utilisateur.

Dit autrement, un systeme de recommandation cherche a predire la valorisation ou la preference qu'un utilisateur attribuerait a un objet (livre, musique, film...) ou a un element social (personne, groupe, communaute) qu'il n'avait pas encore considere.

Par ailleurs, les systemes de recommandation sont des systemes d'aide a la decision permettant de proposer des objets susceptibles d'interesser l'utilisateur afin de l'aider dans sa prise de decision.

Un systeme de recommandation requiert generalement 3 etapes:

1. La premiere consiste a recueillir de l'information sur l'utilisateur.
2. La deuxieme consiste a batir une matrice ou un modele utilisateur contenant l'information recueillie.
3. La troisieme consiste a extraire a partir de cette matrice une liste de recommandations.

3.1 Collecte d'Information

Pour etre pertinent, un systeme de recommandation doit pouvoir faire des predictions sur les interets des utilisateurs. Il faut donc pouvoir collecter un certain nombre de donnees sur ceux-ci afin d'etre capable de construire un profil pour chaque utilisateur.

Une distinction peut etre faite entre 2 formes de collecte de donnees:

3.1.1 Collecte de donnees explicites – Filtrage actif

Repose sur le fait que l'utilisateur indique explicitement au systeme ses interets.

- **Exemple:** demander a un utilisateur de commenter, taguer/etiqueter, noter, aimer ou encore ajouter comme favoris des contenus (objets, articles...) qui l'interessent. On utilise souvent une echelle de *ratings* allant de 1 etoile (je n'aime pas du tout) a 5 etoiles (j'aime beaucoup) qui sont ensuite transformees en valeurs numeriques afin de pouvoir etre utilisees par les algorithmes de recommandation.

- **Avantage:** capacité à reconstruire l'historique d'un individu et capacité à éviter d'agréger une information qui ne correspond pas à cet unique utilisateur (plusieurs personnes sur un même poste).
- **Inconvénient:** les informations recueillies peuvent contenir un biais dit de déclaration. Il s'agit de l'écart entre la donnée déclarée et celle mesurée

3.1.2 Collecte de données implicite – Filtrage passif

La collecte des données implicite repose sur une observation et une analyse des comportements de l'utilisateur, effectuées de façon implicite dans l'application qui embarque le système de recommandation, le tout se fait en "arrière-plan" (sans rien demander à l'utilisateur).

- **Exemples:**
 - Obtenir la liste des éléments que l'utilisateur a écoutés, regardés ou achetés en ligne.
 - Analyser la fréquence de consultation d'un contenu par un utilisateur, le temps passé sur une page.
 - Monitorer le comportement en ligne de l'utilisateur.
 - Analyser son réseau social.
- **Avantage:** Aucune information n'est demandée aux utilisateurs, toutes les informations sont collectées automatiquement. Les données récupérées sont *a priori* justes et ne contiennent pas de biais de déclaration.
- **Inconvénient:** Les données récupérées sont plus difficilement attribuables à un utilisateur et peuvent donc contenir des biais d'attribution (utilisation commune d'un même compte par plusieurs utilisateurs). Un utilisateur peut ne pas aimer certains livres qu'il a achetés, ou il peut l'avoir acheté pour quelqu'un d'autre.

3.2 Profil Utilisateur

Dans un système d'aide à la décision, l'utilisateur est défini par l'ensemble de ses préférences et par des données récoltées par le système. Ces données sont stockées dans ce que l'on appelle un profil utilisateur qui peut contenir :

- des données personnelles de l'utilisateur comme son nom, son sexe, son adresse.
- les préférences de l'utilisateur par rapport à un problème de décision.

Systemes de recommandation

- un historique des actions de l'utilisateur sur le système ou pour un site internet l'historique des pages visitées.

Un autre point important est comment le temps influence le profil de l'utilisateur. Les intérêts des utilisateurs, généralement, évoluent au cours du temps. Les données du modèle utilisateurs devraient donc constamment être réajustées pour rester conformes aux nouveaux centres d'intérêts de l'utilisateur.

3.3 Liste de recommandations

Pour extraire une liste de suggestions à partir d'un modèle utilisateur, les algorithmes utilisent la notion de mesure de similarité entre objets ou personnes décrits par le modèle utilisateur. La similarité a pour but de donner une valeur ou un nombre (au sens mathématique du terme) à la ressemblance entre 2 choses. Plus la ressemblance est forte, plus la valeur de la similarité sera grande. A l'inverse, plus la ressemblance est faible, et plus la valeur de la similarité sera petite. On verra quelques exemples par la suite.

3.4 Types de système de recommandation

Il existe plusieurs façons de faire des recommandations à l'utilisateur. Nous allons présenter dans les sections suivantes les quatre méthodes utilisées en aide à la décision, la recommandation personnalisée qui se base sur le comportement passé de l'utilisateur, la recommandation sur le contenu qui elle se base sur les caractéristiques des objets, la recommandation collaborative qui se base sur l'ensemble des utilisateurs et enfin la recommandation hybride qui comme son nom l'indique mixe les trois premières méthodes.

3.4.1 Recommandation Personnalisée

Il s'agit de recommander des objets sur la base du comportement passé de l'utilisateur.

Exemples:

- Des produits achetés ou sélectionnés sur un site de e-commerce, ainsi qu'un certain nombre d'actions ou décisions effectuées par l'utilisateur qui permettent de prédire de nouveaux produits susceptibles de l'intéresser.

- Les annonces publicitaires (par ex. AdSense de Google) sont considérées comme des systèmes de recommandation personnalisée qui se basent sur le comportement passé de l'utilisateur (navigation, clics, historique de recherche...)

3.4.2 Recommandation par le contenu

Il s'agit de recommander des objets (ou contenus) en se basant sur les qualités et propriétés intrinsèques de l'objet lui-même et en les corrélant avec les préférences et intérêts de l'utilisateur. Ce type de système va donc extraire un certain nombre de caractéristiques et attributs propres à un contenu, afin de pouvoir recommander à l'utilisateur des contenus additionnels possédant des propriétés similaires. Cette méthode crée un profil pour chaque objet ou contenu, c'est-à-dire un ensemble d'attributs/propriétés qui caractérisent l'objet. Des exemples de systèmes utilisant ce principe peuvent être trouvés dans [121], [122].

Exemples:

- La décision de sélection et recommandation ou non d'un document à un utilisateur peut se baser sur le contenu (les mots présents) dans celui-ci, c'est-à-dire sur une comparaison des thèmes abordés dans le document par rapport aux thèmes qui intéressent l'utilisateur.
- Pour décider la suggestion d'un article (news) ou non, un système de recommandation peut se baser sur les mots-clés principaux de l'article et les comparer avec les mots-clés apparaissant dans d'autres articles que l'utilisateur a évalués positivement dans le passé.
- Dans le cas d'un site de vente de livre en ligne, on va se baser sur les caractéristiques du livre pour effectuer des recommandations, comme par exemple le sujet que traite l'ouvrage, son genre, son auteur, l'éditeur, etc. On pourrait ainsi recommander le livre Harry Potter à un utilisateur, si on sait d'une part que ce livre est un roman fantastique et d'autre part que l'utilisateur aime les romans fantastiques. Un système de recommandation pourra donc accomplir cette tâche seulement s'il a à disposition 2 types d'information: 1) la description des caractéristiques du livre et 2) un profil utilisateur qui décrit les intérêts (passés) de celui-ci en termes de préférence de type de livre. La tâche de recommandation consiste donc à déterminer les livres qui correspondent le mieux aux préférences de l'utilisateur.

Profil des livres

Title	Genre	Author	Type	Price	Keywords
<i>The Night of the Gun</i>	Memoir	David Carr	Paperback	29.90	press and journalism, drug addiction, personal memoirs, New York
<i>The Lace Reader</i>	Fiction, Mystery	Brunonia Barry	Hardcover	49.90	American contemporary fiction, detective, historical
<i>Into the Fire</i>	Romance, Suspense	Suzanne Brockmann	Hardcover	45.90	American fiction, murder, neo-Nazism
...					

Profil/Préférences Utilisateur

Title	Genre	Author	Type	Price	Keywords
...	Fiction, Suspense	Brunonia Barry, Ken Follett	Paperback	25.65	detective, murder, New York

Les algorithmes de recommandation par le contenu permettent de développer des modèles afin de trouver des patterns ou motifs semblables entre différentes données. Ils évaluent à quel point un contenu pas encore vu par l'utilisateur est similaire aux contenus que celui-ci a évalués positivement dans le passé. Pour ce faire, on utilise la notion de similarité qui peut être mesurée de plusieurs manières:

- Le système peut tout simplement vérifier si le livre se trouve dans la liste des genres préférés de l'utilisateur. Dans ce cas la similarité sera de 0 ou 1 (binaire/booléen).
- Une autre façon serait de ne pas se baser sur le genre du livre, mais sur les mots-clés qui caractérisent l'ouvrage, et calculer la similarité de chevauchement entre les mots-clés du livre qui va éventuellement être suggéré avec les mots-clés préférés de l'utilisateur. Un des indicateurs de mesure de similarité qu'on utilise dans le cas d'un objet avec des propriétés multi-valeurs (cas des mots-clés) est le coefficient de Dice [123]. Si chaque livre est décrit par un ensemble de mots-clés, permet de mesurer le degré de similarité entre 2 livres à partir de leurs mots-clés.

le coefficient de Dice

$$s = \frac{2|X \cap Y|}{|X| + |Y|} \quad (55)$$

où, $|X|$ est le nombre d'éléments de X . L'indice peut varier de 0 (quand X et Y sont disjoints) à 1 (quand X et Y sont égaux).

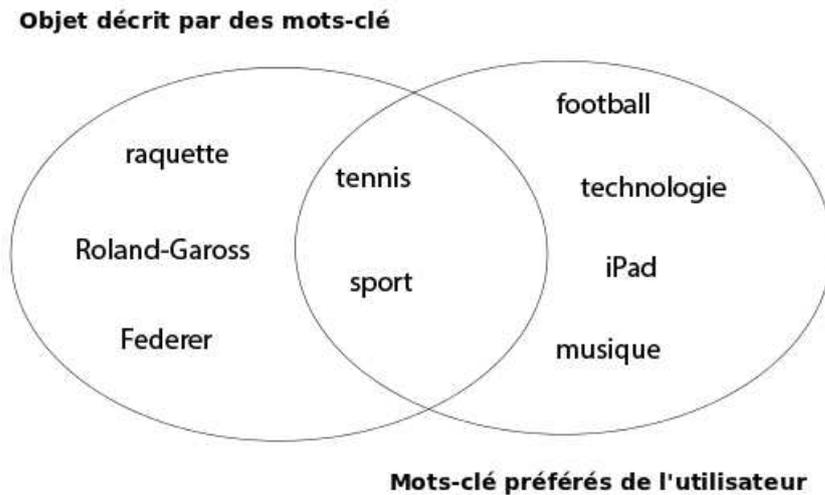


Figure 2 : Mesure de similarité

Dans le cas de documents de texte, on peut également imaginer que le contenu de chaque document soit décrit à partir d'une liste de tous les mots apparaissant dans le document, et décrire chaque document par un vecteur de valeurs binaires/booléennes, où un 1 indiquerait que le mot apparaît dans le document et un 0 indiquerait que le mot n'apparaît pas dans le document. Si le profil utilisateur est décrit par une liste similaire (un 1 indique un intérêt pour un mot-clé), on peut facilement associer et proposer des documents en mesurant le chevauchement entre les mots-clés qui intéressent l'utilisateur et les mots contenus dans le document. Le problème qui surgit avec une telle approche est qu'on assume que chaque mot a la même importance au sein d'un document. Or un mot qui apparaît très souvent ne caractérise pas forcément mieux un document. En plus, un plus large chevauchement entre les préférences de l'utilisateur et un document apparaîtra naturellement plus le document est long. En conséquence, le système de recommandation tendra à proposer des longs documents. Pour résoudre ces problèmes, on applique des techniques qui permettent de réduire le poids des mots qui apparaissent très souvent et d'augmenter le poids de ceux qui apparaissent plus rarement. Pour rendre encore plus compacte la liste des mots présents dans un document, on peut aussi ignorer un certain nombre de mots qui sont sans intérêt, comme les prépositions,

les articles, etc. En normalisant aussi différentes variantes d'un même mot, par exemple les verbes conjugués sont remplacés par leur infinitif (volait par voler). On peut aussi décider de limiter la liste aux n mots les plus pertinents apparaissant dans les documents. Différents travaux ont montré que le nombre optimum de mots à prendre en compte se trouve aux alentours de 100. Si trop peu de mots sont sélectionnés (moins de 50), on voit que d'importantes caractéristiques d'un document ne sont pas couvertes. Et si trop de mots sont inclus, certains ont peu d'importance et apportent plus de bruit qu'autre chose, ce qui réduit la qualité de la recommandation. On utilise aussi des bases de connaissances lexicales externes pour éliminer des mots non pertinents.

De façon générale, les algorithmes de recommandation par le contenu assignent un poids plus ou moins élevé aux attributs d'un contenu selon l'importance de ceux-ci. Le poids exprime donc l'importance de chaque attribut pour un utilisateur.

Un autre algorithme de recommandation par le contenu couramment utilisé est la méthode de retour de pertinence de Rocchio (Rocchio's relevance feedback) [124] ou classification de Rocchio. L'idée de cette méthode est de prendre en compte le point de vue des utilisateurs sur les recommandations proposées. Le principe, simple, repose sur le fait que l'utilisateur est le seul à savoir exactement ce qu'il recherche et apprécie, il est donc le plus à même de juger de la pertinence des suggestions retournées par un système de recommandation. Partant de cette idée, les systèmes exploitant le retour de pertinence permettent de récupérer ces « jugements » utilisateurs et les exploitent pour améliorer et affiner les recommandations. Techniquement, l'idée est dans un premier temps de décomposer les contenus évalués par l'utilisateur en 2 groupes (vecteurs, clusters) D^+ (contenus évalués positivement et pertinents) et D^- (contenus évalués négativement et non pertinents). Par un procédé mathématique on calcule une moyenne (prototype) qui permet de déterminer le centre (centroid) de chacun des 2 groupes (vecteurs) D^+ et D^- . Lorsqu'une nouvelle requête est effectuée par l'utilisateur, les résultats à recommander sont pondérés en fonction des 2 groupes D^+ et D^- . C'est-à-dire, on ajoute au résultat de la requête la liste D^+ et on soustrait (retire) la liste D^- , le tout de façon pondérée. L'effet obtenu est que les résultats retournés se rapprochent à chaque itération un peu plus du centre (centroid) de la liste D^+ des contenus évalués positivement par l'utilisateur. Avec cette technique, on voit qu'il est possible d'affiner la requête initiale des utilisateurs au fur et à mesure que ceux-ci fournissent des jugements de pertinence sur des documents consultés. Le système de recommandation est donc doté d'une boucle de retour qui permet au même système de reformuler des recommandations plus pertinente en favorisant ou écartant certains contenus.

Avantages et inconvenients

Avantage: ce type de recommandation par le contenu n'a pas besoin d'une large communauté d'utilisateurs pour pouvoir effectuer des recommandations. Une liste de recommandations peut être générée même s'il n'y a qu'un seul utilisateur.

Inconvénients: certains attributs peuvent être extraits automatiquement d'un contenu, comme par exemple pour un document de texte, on peut relativement facilement déduire certaines propriétés sémantiques automatiquement. On voit aussi qu'en pratique, des caractéristiques et propriétés comme le genre d'un livre ou la liste des acteurs dans un film, sont généralement fournis par les créateurs du livre ou du film et sont généralement offerts sous forme électronique. Ce qui reste plus problématique, c'est l'acquisition de caractéristiques subjectives et qualitatives... Des propriétés, comme par exemple le style, le design... peuvent difficilement être acquises automatiquement, et devront plutôt être introduites manuellement avec tout ce que ça implique, comme le coût, les éventuelles erreurs...

3.4.3 Recommandation Sociale

Appelée aussi recommandation collaborative [125] [126], consiste à recommander des choses sur la base du comportement passé des utilisateurs similaires, en effectuant une corrélation entre des utilisateurs ayant des préférences et intérêts similaires. On utilise des méthodes qui collectent et analysent des données sur le comportement, les activités, les préférences des utilisateurs et des algorithmes tentent de prédire ce que l'utilisateur aimera en cherchant des utilisateurs qui ont les mêmes comportements que l'utilisateur à qui l'on souhaite faire des recommandations. L'idée sous-jacente est de dire que si une personne A a la même opinion (ou les mêmes goûts) qu'une personne B sur un objet x , alors la personne A a plus de chance d'avoir la même opinion que B sur un autre objet y , plutôt que d'avoir la même opinion que quelqu'un choisi au hasard pour l'objet y . L'idée de base est donc de dire que si des utilisateurs ont partagé des mêmes intérêts dans le passé, il y a de fortes chances qu'ils partagent aussi les mêmes goûts dans le futur.

Les techniques de recommandation sociale (filtrage collaboratif) sont classées en 2 sous-types:

- **memory-based**
- **model-based**

User-based nearest neighbor (user-centric / memory-based)

Les algorithmes de recommandation sociale utilisent généralement différentes variantes d'un mécanisme basé sur le voisinage proche. Dans cette approche, un nombre d'utilisateurs est identifié et sélectionné sur la base de la similarité de leurs intérêts et préférences avec l'utilisateur actif. On utilise alors principalement les *ratings* (par ex: films) de ces utilisateurs "voisins" pour calculer des similarités avec l'utilisateur actif. Pour chaque produit *p* que l'utilisateur n'a pas encore vu, une prédiction est faite en se basant sur les ratings de *p* assignés par le panel d'utilisateurs voisins. Cette méthode suppose 2 conditions initiales (inconvenients):

1. On assume que si des utilisateurs ont eu des goûts similaires dans le passé, ils auront aussi des goûts similaires dans le futur.
2. Les préférences des utilisateurs restent stables et cohérentes dans le temps.

Pour bien comprendre, on peut imaginer un tableau ou matrice de *ratings* de films, avec sur un axe les utilisateurs et sur un autre les films. Chaque cellule de la matrice contient le rating donné par un utilisateur pour un film. Un signe "+" indique que l'utilisateur a aimé le film, un signe "-" qu'il ne l'a pas aimé, pas de signe signifie qu'il n'a pas d'avis particulier sur ce film. Pour pouvoir prédire si Ken apprécierait le film "Fargo" et éventuellement lui recommander ce film, on compare les ratings de Ken à ceux des autres utilisateurs sélectionnés. On peut alors voir que Ken et Mike ont des ratings identiques, et que Mike a aimé le film Fargo, on pourrait alors prédire que Ken aimera aussi ce film et lui faire cette suggestion (user-centric).

Movie ratings

	Amy	Jef	Mike	Chris	Ken
The Piano	-	-	+		+
Pulp Fiction	-	+	+	-	+
Clueless	+		-	+	-
Cliffhanger	-	-	+	-	+
Fargo	-	+	+	-	?

Au lieu de se baser uniquement sur l'utilisateur le plus semblable, la prédiction est normalement calculée à partir de la moyenne pondérée des ratings de plusieurs utilisateurs. Le poids donné au rating de chaque utilisateur est déterminé par le degré de corrélation entre cet utilisateur et l'utilisateur pour qui on désire faire la recommandation. Pour mesurer le degré de corrélation entre 2 utilisateurs, on fait appel généralement au coefficient de corrélation de Pearson [127]. D'autres indicateurs de mesure de la similarité ou proximité entre utilisateurs sont aussi utilisés comme la corrélation de Ringo, ou le coefficient de corrélation de Spearman [128]. Le coefficient de corrélation de Pearson se calcule à partir des valeurs présentes dans la matrice et peut prendre une valeur allant de +1 (forte corrélation positive) à -1 (forte corrélation négative). Par exemple, un coefficient de corrélation 0.70 entre 2 utilisateurs indiquerait une bonne similarité de leurs intérêts. On introduit aussi souvent dans le calcul de la mesure de Pearson un facteur de pondération afin de prendre en compte et pondérer le fait que certains utilisateurs ont toujours tendance à donner seulement de bons ratings, ou que d'autres ne vont jamais donner le rating maximum à un produit.

Les systèmes de recommandation doivent généralement aussi gérer un grand nombre d'utilisateurs. Faire des recommandations à partir des ratings de millions d'utilisateurs peut avoir de sérieuses implications en termes de performance. Ainsi, quand le nombre d'utilisateurs atteint un certain seuil, une sélection des "meilleurs" voisins doit être faite. Pour déterminer quels sont les voisins les plus pertinents à sélectionner, on utilise généralement l'algorithme du *k*-nearest neighbor (*k*-NN) qui permet de sélectionner seulement les *k* meilleurs voisins ayant la plus haute valeur de corrélation. Une autre approche serait de sélectionner seulement les voisins possédant une corrélation plus grande qu'un certain seuil. Le problème dans ces deux approches réside dans le fait de sélectionner le bon nombre *k* d'utilisateurs voisins à prendre en compte ou de déterminer la bonne valeur du seuil de corrélation. Si le nombre *k* est trop petit ou si le seuil de similarité est trop élevé, le nombre d'utilisateurs voisins pris en compte sera trop réduit pour effectuer une bonne prédiction (problème du reduced coverage/couverture réduite). Dans le cas contraire, lorsque que *k* est trop élevé ou que le seuil est trop faible, l'ensemble du panel de voisins est trop large (trop d'utilisateurs avec un degré de similarité limité), ce qui amène du bruit additionnel dans les prédictions. Plusieurs travaux et situations réelles ont montré qu'un panel composé de 20 à 50 voisins donne de bons résultats.

On n'est pas non plus limité aux ratings ou like/dislike, on peut aussi se baser sur des données plus implicites en observant le comportement de l'utilisateur sur le site, par opposition à la récolte de données explicite comme l'est le rating. On peut observer par exemple quelle

musique il a écouté, quel article il a lu, et on croise ces infos avec celles du reste des utilisateurs afin de lui proposer de nouvelles suggestions.

La recommandation sociale *user-centric* a cependant ses limites. Lorsqu'il s'agit d'un gros site qui gère des millions d'utilisateurs et un catalogue de milliers de produits, il faut scanner un grand nombre de voisins potentiels, ce qui rend impossible la recommandation en temps réel. Pour pallier ce problème, les gros sites implémentent souvent une technique différente plus apte au traitement préalable des données hors-ligne (offline preprocessing), la recommandation sociale *item-centric* (à ne pas confondre avec la recommandation objet).

Item-based nearest neighbor (item-centric / model-based)

Cette autre approche, appelée aussi item-to-item, propose une inversion de l'approche user-based nearest neighbor. Au lieu de mesurer la corrélation entre des utilisateurs, les *ratings* sont utilisés pour mesurer la corrélation entre les contenus (films), en s'aidant toujours du coefficient de corrélation de Pearson, mais cette fois-ci appliqué au contenu. Si par exemple les *ratings* des deux films "Fargo" et "Pulp Fiction" ont une parfaite corrélation, c'est-à-dire qu'ils ont reçu les mêmes ratings (positif, négatif ou nul) de la part des utilisateurs, on peut ainsi prédire que Ken aimera le film "Fargo" car il a aimé "Pulp Fiction" (les ratings des utilisateurs sur ces deux films sont parfaitement corrélés).

Pour le dire autrement, l'approche item-centric propose de rechercher en premier lieu des contenus similaires et ensuite de faire une recommandation à l'utilisateur. Cette approche permet de faire un traitement préalable sur la matrice pour déterminer les contenus similaires et ainsi pouvoir effectuer des prédictions en temps réel, contrairement à l'approche user-centric très gourmande en mémoire.

Autant le coefficient de corrélation de Pearson est très utilisé pour déterminer des utilisateurs similaires (cas de la recommandation sociale user-centric), dans l'approche item-centric, on utilise plutôt comme indicateur de mesure de similarité entre items un autre indicateur qu'on appelle de similarité cosinus ajustée. Les valeurs possibles pour cet indicateur vont comme dans le cas de mesure de Pearson de +1 (forte similarité positive) à -1 (forte similarité négative). Une fois que la similarité entre les items a été établie à l'aide de la similarité cosinus ajustée, on peut alors prédire un rating pour un item grâce aux ratings effectués par l'utilisateur sur les items similaires.

Dans l'approche item-centric, l'idée est donc de construire à l'avance la matrice de similarité entre items. Et en temps réel, on peut facilement ensuite déduire la prédiction de

Systemes de recommandation

recommandation (le *rating*) d'un produit pour un utilisateur actif en déterminant depuis la matrice déjà construite quels sont les produits les plus similaires, et en calculant la valeur moyenne de ratings sur ces produits effectués par les utilisateurs voisins.

Avantages et inconvénients

Avantage: L'approche purement de recommandation sociale n'exploite pas ou ne demande aucune connaissance sur les contenus eux-mêmes. Par exemple, dans le cas d'un magasin de vente de livres en ligne, le système de recommandation collaboratif n'a pas besoin de savoir le type de contenu du livre, son genre, qui en est l'auteur, etc... Pas besoin de se baser sur l'analyse des propriétés intrinsèques d'un livre ou d'un contenu, la recommandation sociale est capable de recommander des contenus sans avoir besoin de comprendre le sens ou la sémantique du contenu lui-même. Les informations propres au livre n'ont pas besoin d'être introduite dans le système.

Inconvénients:

- **Scalability:** souvent, les plateformes sur lesquelles sont utilisés les filtres collaboratifs ont des millions d'utilisateurs, de produits et contenus. Cela demande donc beaucoup de puissance de calcul pour pouvoir proposer des suggestions aux utilisateurs. La recommandation sociale de type user-centric est aussi appelée *memory-based*, car la base de données des ratings est maintenue en permanence en mémoire dans le serveur et utilisée directement pour générer des recommandations à l'utilisateur actif. Bien que cette approche *memory-based* soit théoriquement plus précise, car elle a à sa disposition en permanence et en temps réel toutes les données pour générer les recommandations, elle souffre de problèmes de *scalability* pour des bases de données de millions d'utilisateurs et de millions de contenus. Dans une approche item-centric ou *model-based*, les données sont préalablement traitées hors-ligne. Ensuite lors de l'exécution de l'application ou service web, le modèle "appris" ou pré-traité sera utilisé pour effectuer les prédictions. L'approche *model-based* permet d'éviter le problème de *scalability*.
- **Cold Start:** les systèmes de recommandation sociale ont besoin de beaucoup de données et beaucoup d'utilisateurs pour être performants. Le lancement d'un service de recommandation peut souffrir au début du manque d'utilisateurs et d'informations sur ceux-ci.

- **Sparsity** (Rareté): le nombre de produits ou contenus est énorme sur certaines plateformes, et même les utilisateurs les plus actifs auront noté ou valorisé qu'un tout petit sous-ensemble de toute la base de données. Donc, même l'article le plus populaire n'aura que très peu de bonnes notes. Dans une telle situation, deux utilisateurs auront peu d'articles valorisés en commun, ce qui rend plus difficile la tâche de corrélation. C'est une situation qu'on retrouve lorsque le système dispose d'un ratio élevé de contenu par rapport aux utilisateurs, et qu'on retrouve aussi souvent au stade initial du lancement du service de recommandation, ce qui nous ramène au problème de cold start (qui peut être vu comme un cas spécial du problème de rareté).

3.4.4 Recommandation Hybride

Une combinaison des trois approches ci-dessus. Les méthodes hybrides [129] [130] sont de plus en plus utilisées, car elles permettent de résoudre des problèmes comme le *cold start* et la *sparsity* (rareté) qu'on retrouve dans une approche de recommandation uniquement sociale. D'autre part, si par exemple on considère 2 utilisateurs avec les mêmes goûts mais qui n'ont pas évalué ou "raté" des objets en commun, un filtrage collaboratif pur ne les considérera pas comme similaires ou voisins. Rappelons que la mesure de similarité standard ne prend en compte que les éléments pour lesquels l'utilisateur actif et l'utilisateur à comparer ont effectué un rating. L'idée est alors de pouvoir assigner une valeur par défaut aux éléments qui ont été "ratés" seulement par un des deux utilisateurs, afin d'améliorer la qualité de prédiction en cas de rareté (sparsity). En appliquant préalablement un algorithme de recommandation objet sur les contenus pour en exploiter leurs descriptions et caractéristiques, accompagné ensuite d'un algorithme de recommandation sociale pour effectuer les recommandations peut aider à résoudre ces limitations. Autrement dit, pour les cas de rareté (sparsity), lorsque peu d'items ont été évalués par les utilisateurs et qu'un filtrage collaboratif n'est pas possible, ce qu'on fait, c'est qu'on assigne en premier lieu un pseudo-rating ou vote artificiel par défaut à l'utilisateur sur les contenus disponibles en utilisant préalablement un algorithme objet, puis on applique ensuite sur la matrice (contenant peu de vrais rating et beaucoup de pseudo-ratings) un filtrage collaboratif.

Exemples:

Amazon utilise les 3 approches (personnalisée, sociale et objet). Amazon [131] possède un système très sophistiqué, les recommandations sont d'une part personnalisées en se basant sur le comportement individuel passé de l'utilisateur (historique de navigation et historique

Systemes de recommandation

d'achat), et d'autre part Amazon utilise aussi les caracteristiques de l'article lui-meme (recommandation objet) et les comportements d'autres personnes (recommandation sociale). Tous ceux qui ont deja fait un achat sur Amazon ont probablement du lire le message de la part d'Amazon qui nous dit "les gens qui ont achete x ont aussi achete y". Cette approche est tout simplement l'approche item-based nearest neighbor (item-centric ou item-based collaborative filtering) [132] qu'on a decrit auparavant. Il semblerait que le systeme de recommandation d'Amazon represente plus du 30% du CA global de la societe (en 2009)!



Figure 3 : Systeme de recommandation d'Amazon

Netflix est aussi un bon exemple de systeme hybride. Il propose des recommandations en comparant les habitudes de visionnage des films d'utilisateurs similaires (recommandation sociale) et suggere aussi des films qui partagent des caracteristiques avec des films que l'utilisateur a note positivement (recommandation objet).

Google, lui se focalise aussi sur une combinaison de 3 approches pour ameliorer son produit phare qui est son moteur de recherche:

- **Recommandation Personnalisee:**
 - Google customise nos resultats de recherche, quand cela est possible, en se basant sur notre localisation et/ou nos dernieres recherches.
 - Lorsqu'on est connecte a notre compte Google, il propose un contenu encore plus pertinent en fonction de notre historique de recherche.
- **Recommandation Sociale:**

Systemes de recommandation

- L'algorithme du PageRank est de manière intrinsèque un outil basé sur de la recommandation sociale dans la mesure où il utilise les liens entre les pages web.
- L'utilisation de contenus provenant de nos cercles de Google+ est aussi une forme de recommandation sociale.
- **Recommandation Objet**
 - Google utilise aussi une approche sémantique pour sa fonction « Did you mean » de son moteur de recherche.

Pandora vs Last.fm

La différence entre les approches de recommandation objet et de recommandation sociale (collaborative) peut être illustrée en comparant deux services de radio en ligne que sont Pandora (seulement disponible aux USA) et Last.fm.

Pandora utilise principalement une approche de recommandation objet. Le système se base sur les propriétés de la chanson et/ou de l'artiste, c'est à dire il décompose dans un premier temps les morceaux de musique afin de mettre en évidence ses propriétés intrinsèques. Chaque morceau de musique disposerait de plus de 400 attributs, permettant de déterminer une sorte de code génétique de la chanson (Music Genome Project). En comparant les propriétés similaires entre différents morceaux, le système arrive alors à nous proposer de nouvelles chansons à écouter, c'est-à-dire une sorte de station radio qui diffuse de la musique avec des propriétés susceptibles de correspondre à nos goûts. L'utilisateur peut ensuite indiquer s'il aime ou pas le morceau en écoute, et permet ainsi à Pandora d'affiner son filtre de recommandation en privilégiant (ou pas) certains attributs selon l'information venue en retour de l'utilisateur.

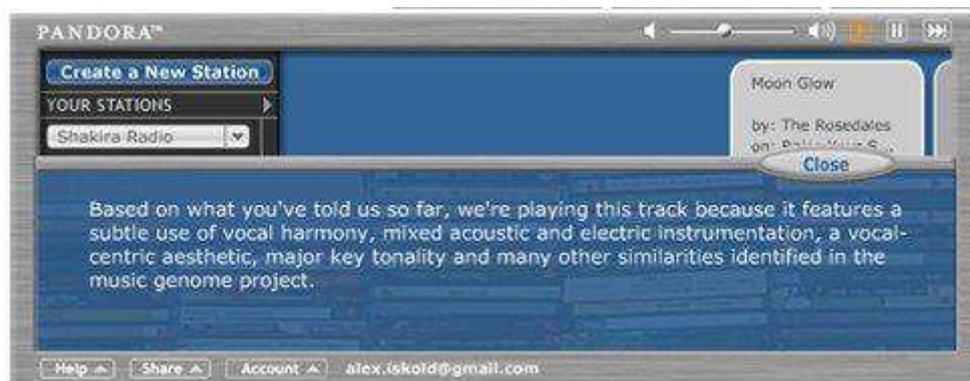


Figure 4 : PANDORA



Last.fm lui utilise un filtre de recommandation sociale (collaborative). Le service crée une station de radio personnalisée en observant quels sont les groupes, artistes et chansons que l'utilisateur a écouté régulièrement et les compare avec les comportements d'écoute d'autres utilisateurs. Last.fm va alors proposer des morceaux fréquemment écoutés par des utilisateurs ayant des goûts similaires.

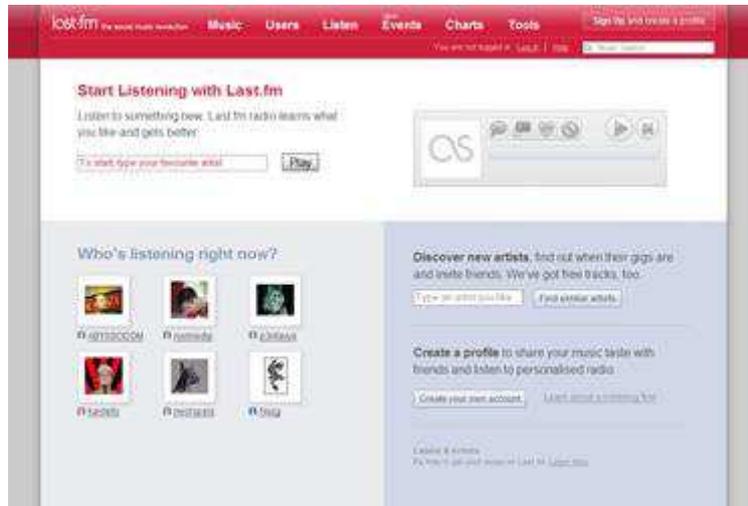


Figure 5 : Last.fm

Chaque type d'approche a ses avantages et inconvénients. En l'occurrence Last.fm requiert beaucoup d'information sur les habitudes d'écoute des utilisateurs afin de produire un système de recommandation efficace (problème du cold start). Pandora lui au contraire a besoin de très peu d'information sur l'utilisateur pour démarrer une radio personnalisée, mais doit disposer d'une base de données gigantesque sur le "génom" de chaque chanson.

Netflix Prize

De 2006 à 2009, Netflix a parrainé un concours, le Netflix prize, offrant un prix de \$1'000'000 à celui qui à partir d'un set de données de plus de 100 millions de *ratings* de films pourrait offrir des recommandations 10% plus précises que celles générées par le système original de la compagnie Netflix. Le 21 septembre 2009, le prix a été attribué à une équipe appelée *BellKor's Pragmatic Chaos*. Les gagnants avaient misé sur un mélange de centaines d'approches algorithmiques et méthodes prédictives qui réunies ensemble permettent d'être plus performant dans la prédiction. La solution la plus performante semble donc bien s'appuyer sur un ensemble de méthodes algorithmiques plutôt que sur le raffinement et l'optimisation d'une seule technique spécifique.

Pour l'anecdote, le Netflix prize n'a pas été renouvelé en 2010, car malgré le fait que le set de million de données fournies aux participants du concours aient été rendues anonymes pour préserver l'identité des clients, deux chercheurs de l'Université du Texas ont été capables

d'identifier des utilisateurs présents dans le set de données utilisé lors du concours en associant et croisant ces données avec des ratings de films soumis sur l'Internet Movie Database (IMDb). Il s'en est suivi qu'en décembre 2009, un des utilisateurs anonymes du set de données a poursuivi Netflix pour violation de la législation américaine en termes de confidentialité.

3.5 Efficacité d'un algorithme de recommandation

La recommandation sociale (filtrage collaboratif) a l'avantage contrairement à la recommandation objet de permettre d'effectuer des prédictions même lorsqu'il y a peu d'information associée au contenu à recommander ou lorsque le contenu est difficile à analyser automatiquement (idées, opinions...). Mais la recommandation sociale souffre aussi d'inconvénients comme le cold start et le problème de rareté (sparsity).

Evaluer l'efficacité des algorithmes de recommandation est loin d'être trivial. En premier lieu, parce que différents algorithmes peuvent être meilleurs ou moins bons en fonction du set de données sélectionné et sur lequel ils sont appliqués. D'autre part, les objectifs fixés par un système de recommandation peuvent être divers et variés. Un système de recommandation peut être mis en place pour estimer avec exactitude la note que donnerait un utilisateur à un élément, alors que d'autres auront comme objectif principal de ne pas proposer des recommandations erronées.

On peut donc légitimement se demander jusqu'à quel point ces différentes méthodes de recommandation sont réellement efficaces. Pour déterminer l'efficacité d'un système, des indicateurs comme la *précision* et le *recall* sont utilisés.

La précision est un indicateur qui représente la qualité de la recommandation, c'est-à-dire à quel point les suggestions proposées sont conformes aux intérêts de l'utilisateur:

$\text{précision} = \text{nombre de suggestions pertinentes} / \text{nombre de suggestions}$

Le recall indique combien de suggestions pertinentes ont été recommandées à l'utilisateur par rapport au nombre total de suggestions pertinentes disponibles dans le système.

$\text{recall} = \text{nombre de suggestions pertinentes proposées à l'utilisateur} / \text{nombre de suggestions pertinentes totale}$

L'indicateur de précision permet de déterminer la probabilité qu'un élément recommandé soit pertinent.

L'indicateur de recall permet de déterminer la probabilité qu'un élément pertinent soit recommandé.

Systemes de recommandation

On utilise aussi des techniques statistiques pour mesurer l'efficacité d'un algorithme de recommandation. L'idée c'est d'évaluer la précision de la prédiction effectuée par le système en comparant les prédictions avec les choix qu'aurait fourni l'utilisateur dans un cas réel. On utilise par exemple l'Erreur Moyenne Absolue (MAE) qui mesure la déviation des recommandations prédites avec les choix réels effectués par les utilisateurs. Plus l'erreur moyenne absolue est faible, meilleure est la prédiction.

Pour conclure, la meilleure mesure de l'efficacité d'un algorithme de recommandation et de la pertinence des suggestions c'est finalement la satisfaction de l'utilisateur, qui n'est pas toujours facile à bien identifier.

3.6 Conclusion

Les systèmes de recommandation reposent sur trois étapes indispensables à savoir la collecte d'information, la définition de profil utilisateur et l'extraction de liste de recommandation. Nous avons discuté les différents types de système de recommandation à savoir la recommandation personnalisée qui consiste à émettre des recommandations sur la base du comportement passé de l'utilisateur, la recommandation par le contenu qui se base sur les caractéristiques intrinsèques des objets, la recommandation collaborative qui exploite l'historique d'autres utilisateurs et enfin la recommandation hybride qui combine les trois premiers types. Nous avons aussi discuté l'efficacité de ces différents types de recommandations.

Chapitre 4: STROMa

STROMa(SysTem of RecOmmendation Multi-criteriA) [102] est un système de recommandation multicritères pour lequel sont mis en œuvre plusieurs opérateurs d'agrégation. L'utilisateur a la possibilité de choisir manuellement un opérateur d'agrégation parmi les opérateurs disponibles, comme il peut se laisser proposer un opérateur d'agrégation par le système lors d'une procédure d'agrégation. L'objectif de STROMa est de proposer un outil d'aide à la décision capable d'adapter sa technique d'agrégation selon le contexte d'utilisation du système. Plusieurs opérateurs d'agrégation multicritères y sont mis en œuvre, notamment : la somme pondérée, l'intégrale de Choquet, MOORA, COPRAS, EDAS, WASPAS, TOPSIS. Pour choisir un opérateur d'agrégation, STROMa utilise un modèle de similarité entre les différents problèmes de décision. Les opérateurs sont également regroupés dans deux grandes catégories à savoir les modèles quantitatifs et les modèles qualitatifs.

Dans ce chapitre, nous présenterons l'architecture de STROMa, l'algorithme détaillé des opérateurs d'agrégation mis en œuvre, ainsi qu'une étude sur comment choisir un opérateur d'agrégation.

4.1 Architecture détaillée de STROMa

STROMa est une application web basée sur une architecture 3 tiers : à savoir le modèle MVC (Modèle Vue Contrôleur). L'utilisateur a la possibilité de définir différents problèmes de décision. Chaque problème de décision est composé d'un ensemble d'éléments, à savoir la description du problème, la description de l'ensemble de ses critères, ainsi que la liste des alternatives.

4.1.1 Problèmes de décision

Que ce soit au niveau de la nation, d'une collectivité locale, d'une administration, d'une entreprise, d'une usine, d'un service, ou tout simplement de la vie familiale, des décisions se prennent quotidiennement en termes de faire ou ne pas faire, faire de telle manière ou de telle autre. Ces décisions peuvent concerner les objectifs de croissance dans un plan de 5 ans, la politique de développement d'une région, la réalisation d'une liaison par train à grande vitesse, un aménagement du barème de l'impôt sur le revenu, une modification

STROMa

du taux des allocations familiales ou encore la localisation d'une nouvelle usine, le lancement d'un nouveau produit, les modalités d'un plan de campagne publicitaire, l'adoption d'un projet de recherche, la politique de distribution des dividendes, l'acceptation des dossiers de crédit, l'achat ou la vente en bourse, la politique d'embauche du personnel, etc. Tout ceux-ci peuvent constituer un problème de décision.

Dans STROMa, un problème de décision est caractérisé par les éléments suivants :

- Code
- Libellé
- Nombre de critères
- Type de problème
- Catégorie de problème
- Liste des critères
- Liste des alternatives
- Liste des préférences
- Utilisateur

Comme pour les critères et les alternatives, le code permet d'identifier un problème de décision de manière unique, il est aussi sur 5 caractères alphanumériques. Le Libellé représente l'intitulé du problème de décision, par exemple « Choix de la voiture du père de famille », « Choix du cuisinier » etc. Le nombre de critères permet la génération de la liste des critères du problème. Le type de problème permet de savoir si les évaluations sont quantitatives ou qualitatives. C'est un paramètre qui permet de sélectionner une catégorie d'opérateur d'agrégation donnée. Les problèmes de décision sont classés dans trois grandes catégories, à savoir les problèmes de choix, de tri et de rangement. L'opérateur d'agrégation ayant le meilleur score dans une catégorie est proposé lors du choix automatique.

La liste des critères est générée en fonction du nombre de critères. Et chaque critère est défini comme expliqué par la suite. De même la liste des alternatives fait partie des composants d'un problème de décision et chaque alternative est décrit en amont.

Les préférences représentent une comparaison entre deux alternatives. Cela peut être une préférence stricte représentée par le symbole P, qui peut se traduire par « au moins aussi bon que » ou une indifférence représentée par le symbole I. L'indifférence indique une sorte

d'équivalence entre deux alternatives. Ces préférences sont surtout utilisées lors de la détermination de la mesure floue de l'intégrale de Choquet. En effet, une modélisation possible des préférences repose sur l'hypothèse suivante: face à deux actions (a_i , a_k), un décideur aura l'une des trois attitudes suivantes: préférer l'une des deux actions par rapport à l'autre, être indifférent entre les deux ou encore refuser de les comparer car, selon lui, elles sont incomparables [133] [134]. Basées sur ce type de réactions, les méthodes dites de surclassement de synthèse introduisent quatre relations de préférence élémentaires [95] [135]:

1. $a_i I a_k$ (situation d'indifférence): elle correspond à la situation où il y a des raisons claires et positives qui justifient une équivalence entre les deux actions;
2. $a_i P a_k$ (situation de préférence stricte): elle correspond à la situation où il y a des raisons claires et positives qui justifient une préférence significative en faveur de l'une (identifiée) des deux options;
3. $a_i Q a_k$ (situation de préférence faible): elle correspond à la situation où il y a des raisons claires et positives qui infirment une préférence stricte en faveur de l'une des deux actions mais ces raisons sont insuffisantes pour en déduire soit une préférence stricte en faveur de l'autre, soit une indifférence entre ces deux actions. Ces raisons ne permettent donc pas d'isoler l'une des deux situations précédentes comme étant la seule appropriée;
4. $a_i R a_k$ (situation d'incomparabilité): elle correspond à l'absence de raisons claires et positives justifiant l'une des trois situations précédentes.

La préférence stricte P indique que le décideur a des raisons claires pour affirmer sans hésitation que l'action a_i est meilleure que a_k . La préférence faible Q exprime une difficulté de conviction; elle ne se traduit pas nécessairement par une différence faible entre les évaluations. L'indifférence I ne représente pas une hésitation, mais plutôt l'équivalence aux yeux du décideur entre les deux actions. Enfin, l'incomparabilité R représente l'impossibilité de porter un jugement de comparaison entre les deux actions.

L'utilisateur a la possibilité de créer autant de problèmes de décision qu'il souhaite. Les différents problèmes de décision sont indépendants entre eux

Exemple : choix du cuisinier, la description du problème de décision.

Problème de décision : Choosing a cook

Description Performances Préférences Validation

1 2 3 4

1 Description des critères

Description du problème

Code * CC Libellé * Choosing a cook Nombre de critères * 3

Type de problème Quantitative Catégorie Problème de Choix Initialiser

Description des critères

N°	Code	Libellé	Poids	Signe	Unité	Description
1	FL	Frog legs	0.4	+ ▾	N ▾	
2	ST	Steak tartare	0.3	+ ▾	N ▾	
3	SC	Scallops	0.3	+ ▾	N ▾	

Annuler Suivant

Figure 6 : Création d'un problème de décision

4.1.2 Critères

Dans un problème d'aide à la décision, un critère représente un point de vue. Il sert de base à un jugement. Il s'agit des jugements de préférence en relation avec la décision. Dans STROMa un critère est caractérisé par les éléments suivants :

- Code
- Libellé
- Poids
- Sens du critère
- Unité de mesure
- Description

Le code permet d'identifier le critère de manière unique. C'est une donnée alphanumérique ne dépassant pas 5 caractères. Le libellé représente l'intitulé du critère ou tout simplement son nom. Quant au poids, il s'agit du degré d'importance de ce critère attribué par l'utilisateur par rapport au problème de décision. À noter que la somme des poids de l'ensemble des critères d'un problème de décision doit être égale à 1. Le sens du critère peut être positif ou négatif, ce qui signifie respectivement un critère à maximiser ou à minimiser. L'unité de mesure étant facultative, permet de connaître la nature du critère. L'utilisateur a la possibilité de définir les unités dont il a besoin. Cela peut être une simple note ou des unités classiques tels que le kilogramme, le litre etc. Et enfin une description détaillée du critère peut être donnée.

Les critères d'un problème de décision sont définis lors de la création du problème. Après la création du problème l'utilisateur a la possibilité d'ajouter d'autres critères, d'en supprimer ou de modifier les critères existants.

Exemple : choix du cuisinier. On dispose 3 critères décrit comme suit sur la figure suivante

Description des critères

N°	Code	Libellé	Poids	Signe	Unité	Description
1	FL	Frog legs	0.4	+ ▾	N ▾	
2	ST	Steak tartare	0.3	+ ▾	N ▾	
3	SC	Scallops	0.3	+ ▾	N ▾	

Figure 7 : Définition des critères d'un problème de décision

4.1.3 Alternatives

Une alternative souvent appelée Action, comme décrit dans la section (1.3.2), représente une possibilité parmi d'autres qu'a le décideur en face d'un problème de décision. Dans STROMa, une alternative est définie par :

- Code
- Libellé

- Vecteur de performance

Le code permet à l'instar des critères, d'identifier une alternative de manière unique. Il est aussi alphanumérique et ne peut excéder 5 caractères. Le libellé représente le nom de l'alternative. Quant au vecteur de performance, il s'agit de l'évaluation de l'alternative par rapport à chaque critère d'évaluation. L'ensemble des vecteurs de performance des différentes alternatives donne la matrice de performance.

Les alternatives sont créées lors de la définition d'un problème de décision. Il est aussi possible d'ajouter d'autres alternatives après la création du problème de décision. La modification et la suppression d'alternatives existantes est aussi possible.

Exemple : choix du cuisinier, la matrice de performance.

Matrice de performances				
Code ↕	Alternative ↕	FL	ST	SC
A	A	18.0	15.0	19.0
B	B	15.0	18.0	19.0
C	C	15.0	18.0	11.0
D	D	18.0	15.0	11.0

Figure 8 : Description des alternatives dans l'exemple choix du cuisinier

4.1.4 Choix de l'opérateur d'agrégation

Pour effectuer une opération d'agrégation sur un problème de décision, l'utilisateur dispose de deux manières, à savoir le choix manuel et le choix automatique de l'opérateur d'agrégation. Un système de notation implicite et explicite est mis en place s'appuyant sur les notes ou les degrés de satisfactions ci-dessous. La note explicite est attribuée par l'utilisateur après une opération d'agrégation, tandis que la note implicite est attribuée par le système lors du choix manuel d'un opérateur ou lors du choix automatique. Ces différentes notes sont détaillées un peu plus bas.

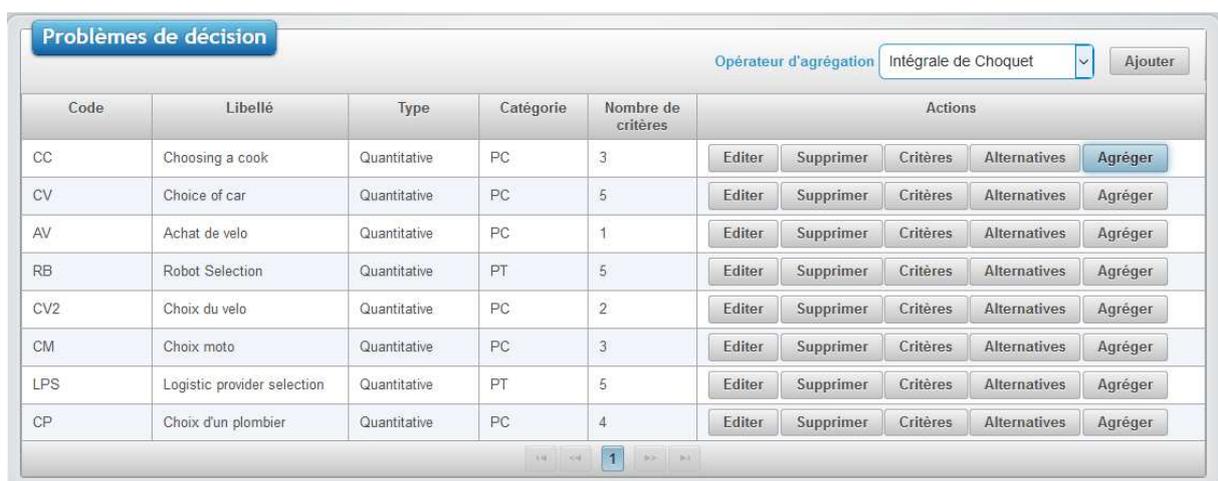
Tableau 9 : Liste des degrés de satisfaction

Code	Libellé	Degré
TM	Très Mauvais	0
MA	Mauvais	1
ME	Médiocre	2
BO	Bon	3
TB	Très bon	4
EX	Excellent	5

4.1.4.1 Choix manuel

Le choix manuel de l'opérateur d'agrégation consiste à ce que l'utilisateur sélectionne lui-même un opérateur d'agrégation dans une procédure d'agrégation. Pour ce faire, il suffit de sélectionner l'opérateur d'agrégation avant de lancer la procédure d'agrégation.

Lorsque l'utilisateur utilise manuellement un opérateur d'agrégation sur un problème de décision, cela provoque de manière implicite l'octroi d'une note. Car ce choix manuel indique une certaine préférence de l'utilisateur vis-à-vis de cet opérateur d'agrégation par rapport à cette catégorie de problème de décision. La note attribuée implicitement est de « 4 » soit « **Très bon** ».

**Figure 9** : Choix manuel d'un opérateur d'agrégation

Après l'agrégation lors du choix manuel, le système calcul le degré de compatibilité de l'opérateur sélectionné par rapport à cette catégorie de problème de décision. Ce degré de compatibilité est calculé en faisant la moyenne de l'ensemble des notes de cet opérateur sur cette catégorie de problème de décision pour cet utilisateur donné.

$$SO_j = \frac{\sum_{i=1}^{nb} N_i}{nb} \quad (56)$$

Où SO_j , est le score de l'opérateur d'agrégation j dans la catégorie de problème donnée. N_i représente la i ème note de l'opérateur d'agrégation j attribuée soit par l'utilisateur de manière explicite soit par le système de manière implicite. nb est le nombre de notes qui ont été attribuées à cet opérateur d'agrégation dans la catégorie de problème donnée par cet utilisateur.

4.1.4.2 Choix automatique

À la différence du choix manuel de l'opérateur d'agrégation, l'utilisateur peut se laisser proposer un opérateur par le système. Pour ce faire, il suffit de sélectionner « Automatique » dans la liste des opérateurs d'agrégation avant de lancer une procédure d'agrégation.

Le système se base sur les différentes notes attribuées soit de manière explicite ou implicite, pour sélectionner l'opérateur d'agrégation ayant le plus grand degré de satisfaction pour cette catégorie de problème de décision. Pour calculer le degré de satisfaction d'un opérateur d'agrégation, le système fait la moyenne de l'ensemble des notes de cet opérateur d'agrégation pour ce type de problème et cette catégorie de problème pour cet utilisateur donné. En tout, trois types de note sont attribués :

1. Lorsque l'utilisateur sélectionne manuellement un opérateur d'agrégation dans une procédure d'agrégation sur un problème de décision, le système attribue de manière implicite une note « Très bon » à cet opérateur pour ce problème. Lorsque l'utilisateur sélectionne un opérateur d'agrégation sur un problème de décision, cela signifie qu'il pense que cet opérateur est adapté à son problème. Il est satisfait du résultat d'une

opération d'agrégation lorsque le classement final proposé correspond à ses préférences, c'est-à-dire au classement qu'il avait en tête.

2. Lorsque le système propose un opérateur d'agrégation de manière automatique, une note égale au score ayant permis de sélectionner cet opérateur est attribuée. En fonction de cette note, l'opérateur est renforcé ou affaibli vis-à-vis de cette catégorie de problème de décision selon que la note soit bonne ou mauvaise.
3. Après chaque procédure d'agrégation, que ce soit manuelle ou automatique, l'utilisateur a la possibilité d'indiquer son degré de satisfaction de manière explicite de cet opérateur vis-à-vis de cette catégorie de problème de décision. Cette note est la plus importante, car l'utilisateur est le seul à savoir s'il est satisfait ou pas après une procédure d'agrégation.

Les notes attribuées ont la structure suivante :

- Utilisateur
- Problème de décision
- Opérateur d'agrégation
- Degré de satisfaction
- Mode d'attribution
- Date d'attribution

L'utilisateur représente le décideur, c'est lui qui effectue l'opération d'agrégation. Il doit disposer d'un compte utilisateur (login et mot de passe) lui permettant d'accéder au système. Le problème de décision doit être créé en amont par l'utilisateur avant de lancer la procédure d'agrégation.

L'opérateur d'agrégation est choisi parmi les opérateurs d'agrégation implémentés à savoir la somme pondérée, l'intégrale de Choquet, MOORA, COPRAS, TOPSIS, EDAS, WASPAS. Toutefois, cette liste d'opérateur est amenée à évoluer en fonction de l'évolution du système.

Après chaque procédure d'agrégation l'utilisateur est appelé à donner son degré de satisfaction vis-à-vis de la procédure d'agrégation par rapport à ce type de problème de décision. Le degré de satisfaction doit être dans la liste décrite ci-dessus. Il exprime la satisfaction de l'utilisateur par rapport à la procédure d'agrégation.

Le mode d'attribution indique le type de notation. Il y a notamment trois types de notation comme expliqué plus haut. Chaque type de notation correspond à un mode d'attribution. Le mode d'attribution manuelle nommé « M » est utilisé lorsque l'utilisateur donne une note explicitement. Tandis que, le mode d'attribution automatique nommé « A » est utilisé lorsque le système propose un opérateur à l'utilisateur. Et enfin, le mode d'attribution « C » est utilisé lorsque l'utilisateur choisi manuellement un opérateur d'agrégation.

The screenshot shows a web interface titled "Problèmes de décision". At the top right, there is a dropdown menu for "Operateur d'agrégation" set to "Automatique" and an "Ajouter" button. Below this is a table with the following data:

Code	Libellé	Type	Catégorie	Nombre de critères	Actions				
CC	Choosing a cook	Quantitative	PC	3	Editer	Supprimer	Critères	Alternatives	Agréger
CV	Choice of car	Quantitative	PC	5	Editer	Supprimer	Critères	Alternatives	Agréger
AV	Achat de velo	Quantitative	PC	1	Editer	Supprimer	Critères	Alternatives	Agréger
RB	Robot Selection	Quantitative	PT	5	Editer	Supprimer	Critères	Alternatives	Agréger
CV2	Choix du velo	Quantitative	PC	2	Editer	Supprimer	Critères	Alternatives	Agréger
CM	Choix moto	Quantitative	PC	3	Editer	Supprimer	Critères	Alternatives	Agréger
LPS	Logistic provider selection	Quantitative	PT	5	Editer	Supprimer	Critères	Alternatives	Agréger
CP	Choix d'un plombier	Quantitative	PC	4	Editer	Supprimer	Critères	Alternatives	Agréger

At the bottom of the table, there are navigation buttons: "1", "<<", ">>", and "1".

Figure 10 : Choix automatique d'un opérateur d'agrégation

4.1.5 Caractéristiques techniques de STROMa

STROMa est une application web, développée en java notamment en JSF (JavaServer Faces). Nous utilisons différents outils dont les plus représentatifs sont les suivants :

- **Mysql** : le SGBD (Système de Gestion de Base de Données) utilisé pour la persistance des données. En effet, l'ensemble des données de l'application sont stockées dans une base de données mysql.
- **JDK** (Java Development Kit) : désigne un ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être compilé, transformé en bytecode destiné à la machine virtuelle Java. C'est l'outil de développement de base en java.
- **Apache tomcat 6** : est un conteneur web libre de servlets et JSP Java EE. Issu du projet Jakarta, c'est un des nombreux projets de l'Apache Software Foundation. Il

implémente les spécifications des servlets et des JSP du Java Community Process, est paramétrable par des fichiers XML et des propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur http.

- **Kappalab**: a été mis à la disposition des utilisateurs à travers le système statistique R [136]. Ce package contient des routines pour la manipulation de divers types de fonctions tels que des capacités. Il peut être utilisé pour calculer plusieurs intégrales non-additives : l'intégrale de Choquet, l'intégrale de Sugeno. Nous l'utilisons dans le cadre de la détermination de la mesure floue de l'intégrale de Choquet.
- **JRI (Java/R Interface)**: est une interface Java / R, qui permet d'exécuter R dans des applications Java en tant que thread unique. Fondamentalement, il charge la bibliothèque dynamique R en Java et fournit une fonctionnalité API Java vers R. Il prend en charge les appels simples aux fonctions R. Il est utilisé dans le cadre de l'utilisation du package Kappalab.
- **PrimeFaces** : est une librairie open source de composants graphiques pour les applications JSF (Java Server Faces) créé par la compagnie Turque PrimeTek. Il s'agit d'un ensemble de plus de 100 composants graphiques.
- **Hibernate** : est une solution open source de type ORM (Object Relational Mapping) qui permet de faciliter le développement de la couche persistance d'une application. Hibernate permet donc de représenter une base de données en objets Java et vice versa.
- Nous utilisons d'autres composants indispensables au fonctionnement d'une application web JSF.

4.2 Les opérateurs d'agrégation mis en œuvre

Dans STROMa plusieurs opérateurs d'agrégation sont mis en œuvre. Chaque opérateur pouvant être sélectionné dans n'importe quel problème de décision, il faut alors une représentation robuste et souple des problèmes de décision et des opérateurs d'agrégation. Chaque problème de décision est caractérisé par un ensemble de critères et un ensemble d'alternatives. Et chaque alternative est évaluée par rapport à ces différents critères, donnant ce que l'on appelle la matrice de performance. Nous allons détailler la mise en œuvre des différents opérateurs d'agrégation mis en œuvre.

4.2.1 Somme pondérée

La somme pondérée comme décrit dans la section 2.32, est relativement facile à mettre en œuvre. Disposant de la liste des critères et de la liste des alternatives avec leur évaluation sur chaque critère, il suffit de suivre l'algorithme suivant pour ce faire :

Données : alternatives[][] matrice composée en ligne par les alternatives et en colonne par leur évaluation sur chaque critère, criteres[] le vecteur poids, nbAlternatives, nbCriteres

Résultats : valeursAgreges[] : un vecteur contenant les valeurs agrégées des différentes alternatives

Début

```

Pour i allant de 1 à nbAlternative faire
    valeursAgreges[i]=0
    Pour j allant de 1 à nbCritere faire
        valeursAgreges[i]=+alternatives[i][j]*criteres[j]
    fin
fin
retourner valeursAgreges
fin

```

4.2.2 Intégrale de Choquet

L'intégrale de Choquet [114] est un opérateur assez complexe, qui permet de prendre en compte les dépendances entre critères. C'est un opérateur qui utilise une mesure floue appelée aussi capacité. La mise en œuvre de l'intégrale de Choquet passe par la détermination de cette mesure floue correspondant aux préférences de l'utilisateur. Afin de pouvoir exploiter de manière simple des intégrales de Choquet, un package nommé « Kappalab » a été mis à la disposition des utilisateurs à travers le système statistique R [136]. Ce package contient des routines pour la manipulation de divers types de fonctions tels que des capacités. Il peut être utilisé pour calculer plusieurs intégrales non-additives : l'intégrale de Choquet, l'intégrale de Sugeno. Une analyse des capacités, en termes de comportement de décision, peut être réalisée par le calcul des différents indices [voir chapitre 2.3.3] tel que la valeur de Shapley, l'indice d'interaction, etc. La célèbre transformation de Möbius, ainsi que d'autres représentations équivalentes de fonctions peuvent également être calculées.

4.2.2.1 Détermination de la mesure floue de l'intégrale de Choquet

Formulation du problème d'identification de capacités

Une fois les fonctions d'utilité commensurables déterminées, il s'agit d'identifier une capacité, si elle existe, telle que l'intégrale de Choquet par rapport à cette capacité représente numériquement les préférences du décideur.

En pratique, le décideur raisonne généralement sur un sous-ensemble O , usuellement de faible cardinal, de l'ensemble X des objets d'intérêts.

Les préférences initiales du décideur, à partir desquelles la capacité doit être déterminée, peuvent prendre la forme :

- d'un pré-ordre partiel \geq_O sur les objets disponibles ;
- d'un pré-ordre partiel \geq_N sur les critères ;
- d'intuitions quantitatives sur l'importance de certains critères ;
- etc.

Dans le contexte considéré, il semble naturel de traduire le pré-ordre partiel \geq_O en utilisant les règles suivantes :

- $a \geq_O b$ peut être traduit par $C_\mu(a) > C_\mu(b)$
- $a \sim_O b$ peut être traduit par $C_\mu(a) = C_\mu(b)$

où μ est la capacité à déterminer.

De même, $i \geq_N j$ peut être traduit par $\Phi_\mu(i) \geq \Phi_\mu(j)$ et $i \sim_N j$ peut être traduit par $\Phi_\mu(i) = \Phi_\mu(j)$.

En traduisant l'ensemble des préférences émises par le décideur, en utilisant les règles ci-dessus, cela aboutit à un problème d'optimisation dont la solution recherchée est la mesure floue μ sur N . Ce problème d'optimisation est en général de la forme suivante :

min ou max $F(\dots)$

$$\text{sous les contraintes suivantes } \left\{ \begin{array}{l} \mu(S \cup i) - \mu(S) \geq 0, \forall i \in N, \forall S \subseteq N \setminus i, \\ \mu(\emptyset) = 0, \mu(N) = 1, \\ C_\mu(a) - C_\mu(b) \geq \delta c, \\ \dots \\ \Phi_\mu(i) - \Phi_\mu(j) \geq \delta sh \\ \dots \end{array} \right. \quad (57)$$

où F est une fonction objectif qui diffère selon la méthode d'identification choisie. Parmi les principales méthodes d'identification de capacités on peut citer:

- Approches fondées sur les moindres carrés [137], [138];
- Approche fondée sur la programmation linéaire [139];
- Moindres carrés généralisés [140];
- Méthode du minimum de variance [141], qui peut également s'interpréter comme une méthode du maximum d'entropie ;
- Méthode du minimum de distance [142].

Nous utilisons le package Kappalab [136] pour la détermination de la capacité à travers la méthode du minimum de variance [141]. D'autres outils tels que JRI (Java\R Interface, permettant d'utiliser un programme R dans java), JDK (Java Development Kit) ainsi que les bibliothèques indispensables au fonctionnement d'une application J2EE sont également utilisés.

Le processus de détermination de la capacité peut se résumer aux étapes suivantes :

1. Définir l'ensemble des critères à l'aide d'un tableau de critères.
2. Définir les performances de chaque alternative pour chacun des critères. Il s'agit du remplissage de la matrice de performance.
3. Etablir un ordre partiel entre les différentes alternatives. Une préférence est définie par un couple d'alternatives et la valeur de préférence. On note deux valeurs de préférences :
 - 1 : la première alternative est préférée à la seconde,
 - 0 : les deux alternatives sont indifférentes ou équivalentes.

Cela permet d'avoir un tableau de préférences définissant l'ordre partiel qui reflète les préférences exprimées par le décideur.

4. Transformer le tableau de préférences en une matrice R contenant toutes les informations préférentielles.
5. Utiliser la fonction **mini.var.capa.ident** du package Kappalab [136] afin de déterminer la capacité correspondante aux informations préférentielles. Cette fonction est exécutée dans la plateforme R en utilisant les composants du JRI. Elle utilise la méthode d'identification du minimum de variance [141]. pour déterminer la capacité.
6. Récupérer la capacité résultante à partir de la plateforme R.

4.2.2.2 Implémentation de l'intégrale de Choquet

Une fois la mesure floue déterminée à l'aide du package Kappalab [136], nous pouvons à présent agréger les performances de chaque alternative afin d'avoir un score global dans l'optique d'établir un classement final. L'intégrale de Choquet est un opérateur d'agrégation assez souple et connu surtout pour sa faculté de prendre en compte les interactions entre les critères.

L'algorithme suivant permet de mettre en œuvre l'intégrale de Choquet.

Données : alternatives[][] matrice composée en ligne par les alternatives et en colonne par leur évaluation sur chaque critère, criteres[] le vecteur poids, mu[] :capacité déterminée en utilisant Kappalab, nbAlternatives, nbCriteres.

Résultats : valeursAgreges[] : un vecteur contenant les valeurs agrégées des différentes alternatives

Début

Pour i allant de 1 à nbAlternative faire

- Permuter le vecteur de performance du plus petit au plus grand. Les critères doivent suivre le même ordre.

valeursAgreges[i]=0

Pour j allant de 1 à nbCritere faire

$a_j = \{j, \dots, N\}$

$a_{jplus} = \{j+1, \dots, N\}$

$valeursAgreges[i] += alternatives[i][j] * (\mu(a_j) - \mu(a_{jplus}))$

fin

fin

retourner valeursAgreges

fin

Nous utilisons l'algèbre de Boole pour déterminer les différents sous-ensembles de critères. Le nombre de sous-ensembles est 2^n , n étant le nombre de critères. Les indices d'interactions ainsi que les valeurs de Shapley sont également calculés une fois la mesure floue déterminée.

4.2.3 MOORA

Comme détaillé dans la section 2.3.5, la méthode MOORA est une méthode MCDA dont l'implémentation peut se décomposer en deux grandes étapes. A savoir la procédure de normalisation et celle de l'agrégation. Pour ce faire, il suffit de suivre les deux algorithmes suivant.

4.2.3.1 Procédure de normalisation

Données : alternatives[][] matrice composée en ligne par les alternatives et en colonne par leur évaluation sur chaque critère, nbAlternatives, nbCrtieres

Résultats : alternativesNormalise[][] : une matrice composée en ligne par les alternatives et en colonne par les valeurs normalisées des évaluations.

Début

```

Pour j allant de 1 à nbCritere faire
    double sommeCarre=0
    Pour i allant de 1 à nbAlternative faire
        sommeCarre +=altrvatives[i][j]* altrvatives[i][j]
    fin
    Pour i allant de 1 à nbAlternative faire
        alternativesNormalise[i][j] =altrvatives[i][j]/sommeCarre
    fin
fin
retourner alternativesNormalise
fin

```

4.2.3.2 Procédure d'agrégation

Après avoir normalisé les évaluations en utilisant l'algorithme ci-dessus, on peut à présent agréger les performances en utilisant l'algorithme suivant :

Données : alternativesNormalise [][] matrice composée en ligne par les alternatives et en colonne par leur évaluation sur chaque critère, criteres[] : le vecteur poids, nbAlternatives, nbCrtieres

STROMa

Résultats : valeursAgreges[] : un vecteur contenant les valeurs agrégées des différentes alternatives

Début

```
Pour j allant de 1 à nbCritere faire
    Pour i allant de 1 à nbAlternative faire
        alternativesNormalise[i][j] = alternativesNormalise [i][j]*critere[j]
    fin
fin
Pour i allant de 1 à nbAlternative faire
    double sommePlus=0
    double sommeMoins=0
    Pour j allant de 1 à nbCritere faire
        si (critere[j] est à maximiser) alors
            sommePlus+=alternativesNormalise[i][j]
        sinon
            sommeMoins+=alternativesNormalise[i][j]
    fin
    valeursAgreges[i]=sommePlus-sommeMoins
fin
retourner valeursAgreges
fin
```

Après la procédure d'agrégation, on peut classer les alternatives en fonction de la valeur agrégée. L'alternative ayant la plus forte valeur agrégée aura le meilleur classement ainsi de suite.

4.2.4 COPRAS

Comme expliqué dans la section 2.3.6, la méthode COPRAS est une méthode MCDA dont l'implémentation peut se décomposer en deux grandes étapes. A savoir la procédure de normalisation et celle de l'agrégation. Pour ce faire, il suffit de suivre les algorithmes suivant :

4.2.4.1 Procédure de normalisation

Données : alternatives[][] matrice composée en ligne par les alternatives et en colonne par leur évaluation sur chaque critère, nbAlternatives, nbCriteres

Résultats : alternativesNormalise[][] : une matrice composée en ligne par les alternatives et en colonne par les valeurs normalisées des évaluations.

Début

```

Pour j allant de 1 à nbCritere faire
    double somme=0
    Pour i allant de 1 à nbAlternative faire
        somme +=alternatives[i][j]
    fin
    Pour i allant de 1 à nbAlternative faire
        alternativesNormalise[i][j] =alternatives[i][j]/somme
    fin
fin
retourner alternativesNormalise
fin

```

4.2.4.2 Procédure d'agrégation

Après avoir normalisé les évaluations en utilisant l'algorithme ci-dessus, on peut à présent agréger les performances en utilisant l'algorithme suivant :

Données : alternativesNormalise [][] matrice composée en ligne par les alternatives et en colonne par leur évaluation sur chaque critère, criteres[] : le vecteur poids, nbAlternatives, nbCriteres

Résultats : valeursAgreges[] : un vecteur contenant les valeurs agrégées des différentes alternatives

Début

```

Pour j allant de 1 à nbCritere faire
    Pour i allant de 1 à nbAlternative faire
        alternativesNormalise[i][j] = alternativesNormalise [i][j]*critere[j]
    fin

```

STROMa

```
fin
Pour i allant de 1 à nbAlternative faire
    double sommePlus=0
    double sommeMoins=0
    Pour j allant de 1 à nbCritere faire
        si (critere[j] est à maximiser) alors
            sommePlus+=alternativesNormalise[i][j]
        sinon
            sommeMoins+=alternativesNormalise[i][j]
    fin
    alternativeSommePlus [i]=sommePlus
    alternativeSommeMoins[i]=sommeMoins
fin
double sommeMoins=0
double somme1Moins=0
Pour i allant de 1 à nbAlternative faire
    sommeMoins+= alternativeSommeMoins[i]
    somme1Moins+=1/ alternativeSommeMoins[i]
fin
Pour i allant de 1 à nbAlternative faire
    valeursAgreges[i]= alternativeSommePlus[i]+(sommeMoins/(
        AlternativeSommeMoins[i]* somme1Moins))
fin
retourner valeursAgreges
fin
```

Après avoir agrégé les performances en utilisant l'algorithme ci-dessus, on peut alors classer les alternatives en fonction de la valeur agrégée. L'alternative ayant la plus forte valeur agrégée aura le meilleur rang ainsi de suite.

4.2.5 EDAS

EDAS est une autre méthode MCDA que nous avons mis en œuvre. Pour plus d'informations sur EDAS voir la section 2.3.9. Cette mise en œuvre se résume à l'algorithme suivant :

Données : alternatives[][] matrice composée en ligne par les alternatives et en colonne par leur évaluation sur chaque critère, criteres[] : le vecteur poids, nbAlternatives, nbCriteres

Résultats : valeursAgreges[] : un vecteur contenant les valeurs agrégées des différentes alternatives

Début

```

Pour j allant de 1 à nbCritere faire
    double somme=0
    Pour i allant de 1 à nbAlternative faire
        somme += alternatives [i][j]
    fin
    double avj=somme/nbAlternatives
    Pour i allant de 1 à nbAlternative faire
        alternativePDA[i][j]=(max(0, alternatives [i][j]-avj)/avj
        alternativeNDA[i][j]=(max(0, avj-alternatives [i][j])/avj
    fin
fin
Pour i allant de 1 à nbAlternative faire
    double SP=0
    double SN=0
    Pour j allant de 1 à nbCritere faire
        SP+= alternativePDA [i][j]*critere[j]
        SN+= alternativeNDA [i][j]*critere[j]
    fin
    alternativeSP[i]=SP
    alternativeSN[i]=SN
fin
double maxPositif= max(alternativeSP[])
double maxNegatif=max(alternativeSN[])

```

STROMa

```
Pour i allant de 1 à nbAlternative faire
    double nsp= alternativeSP[i]/maxPositif
    double nsn=1- (alternativeSN[i]/maxNegatif)
    valeursAgreges[i]=(nsp+nsn)/2
fin
retourner valeursAgreges
fin
```

4.2.6 WASPAS

L'implémentation de la méthode WASPAS se fait en deux phases également, à savoir les phases de normalisation et d'agrégation. Cette méthode est décrite dans la section 2.3.8.

4.2.6.1 Procédure de normalisation

Données : alternatives[][] matrice composée en ligne par les alternatives et en colonne par leur évaluation sur chaque critère, nbAlternatives, nbCriteres

Résultats : alternativesNormalisee[][] : une matrice composée en ligne par les alternatives et en colonne par les valeurs normalisées des évaluations.

Début

```
Pour j allant de 1 à nbCritere faire
    double maxOuMin=alternatives[0][j]
    Pour i allant de 1 à nbAlternative faire
        si (critere[j] est à maximiser) alors
            maxOuMin= (maxOuMin<alternatives[i][j] ? alternatives[i][j] :maxOuMin)
        sinon
            maxOuMin= (maxOuMin>alternatives[i][j] ? alternatives[i][j] :maxOuMin)
    fin
fin
Pour i allant de 1 à nbAlternative faire
    si (critere[j] est à maximiser) alors
        alternativesNormalisee[i][j] = alternatives[i][j]/maxOuMin
    sinon
```

STROMa

```
alternativesNormalisee[i][j] = maxOuMin /alternatives[i][j]
    fin
fin
fin
retourner alternativesNormalise
fin
```

4.2.6.2 Procédure d'agrégation

Après avoir normalisé les évaluations en utilisant l'algorithme ci-dessus, on peut à présent agréger les performances en utilisant l'algorithme suivant :

Données : alternativesNormalise[][] matrice composée en ligne par les alternatives et en colonne par leur évaluation sur chaque critère, criteres[] : le vecteur poids, nbAlternatives, nbCriteres, labda=0.5

Résultats : valeursAgreges[] : un vecteur contenant les valeurs agrégées des différentes alternatives

Début

```
Pour i allant de 1 à nbAlternative faire
    double sommePlus=0
    double sommeMoins=0
    Pour j allant de 1 à nbCritere faire
        sommePlus+=alternativesNormalise[i][j]*criteres[j]
        sommeMoins*=puissance(alternativesNormalise[i][j], criteres[j])
    fin
    valeursAgreges[i]=labda*sommePlus + (1-labda)*sommeMoins
fin
retourner valeursAgreges
fin
```

Après la procédure d'agrégation, on peut classer les alternatives en fonction de la valeur agrégée. L'alternative ayant la plus forte valeur agrégée aura le meilleur classement ainsi de suite.

4.2.7 TOPSIS

Comme détaillé dans la section 2.3.7, la méthode TOPSIS est une autre méthode MCDA dont l'implémentation peut se décomposer en deux étapes. A savoir la procédure de normalisation et celle de l'agrégation. Pour ce faire, il suffit de suivre les deux algorithmes suivant :

4.2.7.1 Procédure de normalisation

Données : alternatives[][] matrice composée en ligne par les alternatives et en colonne par leur évaluation sur chaque critère, nbAlternatives, nbCrtieres

Résultats : alternativesNormalise[][] : une matrice composée en ligne par les alternatives et en colonne par les valeurs normalisées des évaluations.

Début

```

Pour j allant de 1 à nbCritere faire
    double sommeCarre=0
    Pour i allant de 1 à nbAlternative faire
        sommeCarre +=altermatives[i][j]* altermatives[i][j]
    fin
    Pour i allant de 1 à nbAlternative faire
        alternativesNormalise[i][j] =altermatives[i][j]/ sqrt(sommeCarre)
    fin
fin
retourner alternativesNormalise
fin

```

4.2.7.2 Procédure d'agrégation

Après avoir normalisé les évaluations en utilisant l'algorithme ci-dessus, on peut à présent agréger les performances en utilisant l'algorithme suivant :

Données : alternativesNormalise [][] matrice composée en ligne par les alternatives et en colonne par leur évaluation sur chaque critère, criteres[] : le vecteur poids, nbAlternatives, nbCrtieres

Résultats : valeursAgreges[] : un vecteur contenant les valeurs agrégées des différentes alternatives

Début

```

Pour j allant de 1 à nbCritere faire
    Pour i allant de 1 à nbAlternative faire
        alternativesNormalise[i][j] = alternativesNormalise [i][j]*critere[j]
    fin
fin
Pour i allant de 1 à nbCritere faire
    double pis= alternativesNormalise [0][i]
    double nis= alternativesNormalise [0][i]
    Pour j allant de 1 à nbAlternative faire
        si (i == 1) alors
            alternativeSommePlus[j]=0
            alternativeSommeMoins[j]=0
        fin
        si (critere[i] est à maximiser)
            pis=(pis< alternativesNormalise [j][i] ? alternativesNormalise [0][i] :pis)
            nis=(nis> alternativesNormalise [j][i] ? alternativesNormalise [0][i] :nis)
        sinon
            pis=(pis> alternativesNormalise [j][i] ? alternativesNormalise [0][i] :pis)
            nis=(nis< alternativesNormalise [j][i] ? alternativesNormalise [0][i] :nis)
        fin
    fin
    Pour j allant de 1 à nbAlternative faire
        alternativeSommePlus[j]+=pow(alternativesNormalise [j][i] -pis, 2)
        alternativeSommeMoins[j]+=pow(alternativesNormalise [j][i] -nis, 2)
    fin
fin
Pour i allant de 1 nbAlternative faire
    alternativeSommePlus[i]=sqrt(alternativeSommePlus[i])
    alternativeSommeMoins[i]=sqrt(alternativeSommeMoins[i])
    valeursAgreges[i]= alternativeSommeMoins[i]/( alternativeSommeMoins[i]+

```

STROMa

```
        alternativeSommePlus[i])
    fin
    retourner valeursAgreges
fin
```

Après la procédure d'agrégation, on peut classer les alternatives en fonction de la valeur agrégée. L'alternative ayant la plus forte valeur agrégée aura le meilleur classement ainsi de suite.

4.3 Comment choisir un opérateur d'agrégation face à un problème de décision

Il existe de nombreux opérateurs d'agrégation. Ils sont regroupés dans diverses familles, telles que les moyennes, les normes triangulaires et les conormes, Choquet et Sugeno intégrales, et beaucoup d'autres. La question est de savoir comment choisir le plus approprié opérateur d'agrégation pour une application spécifique. Est-ce qu'il faut une seule fonction d'agrégation, ou faut-il utiliser différentes fonctions d'agrégation dans différentes parties de l'application?

Il y a deux composants à la réponse. Tout d'abord, la fonction d'agrégation sélectionnée doit être cohérente avec la sémantique de la procédure d'agrégation. C'est-à-dire, si l'on modélise une conjonction, les fonctions d'agrégation moyenne ou disjonctive ne conviennent pas. La fonction d'agrégation doit-elle être symétrique, avoir un élément neutre ou absorbant, ou être idempotente? Le nombre d'entrées est-il toujours le même? Quelle est l'interprétation des valeurs d'entrée? Répondre à ces questions devrait aboutir à un certain nombre de propriétés mathématiques, sur la base desquelles une classe ou une famille appropriée peut être choisie.

Le deuxième problème consiste à choisir le membre approprié de cette classe ou famille, qui fait ce qu'il est supposé faire, produit des résultats adéquats pour des entrées données. On s'attend à ce que le développeur d'un système ait une idée approximative des résultats appropriés pour certaines entrées de prototypes.

Les données peuvent provenir de différentes sources et sous différentes formes. Premièrement, cela pourrait être le résultat d'une expérience mentale: prenons les valeurs d'entrée (1, 0, 0). Quel résultat attendons-nous?

Deuxièmement, le développeur d'une application peut demander aux experts du domaine de donner leur avis sur les résultats souhaités pour les entrées sélectionnées. Cela peut être fait en présentant aux experts quelques cas prototypiques (soit les vecteurs d'entrée, soit les situations spécifiques au domaine avant qu'elles ne soient traduites dans les entrées). S'il y a plus d'un expert, on peut faire la moyenne de leurs résultats ou traduits dans la plage des valeurs de sortie possible, ou les experts peuvent être amenés à trouver un consensus.

Troisièmement, les données pourraient être collectées dans une expérience, en demandant à un groupe d'experts les valeurs d'entrée et de sortie, mais sans associer ces valeurs à une règle d'agrégation.

Quatrièmement, les données peuvent être collectées automatiquement en observant les réponses des sujets à divers stimuli. Par exemple, en présentant à un utilisateur d'un système informatique des informations et en enregistrant ses actions ou décisions.

Dans le cas le plus typique, les données viennent par paires (x, y) , où $x \in [0, 1]^n$ est le vecteur d'entrée et $y \in [0, 1]$ est la sortie désirée. Il y a plusieurs paires, qui seront notées par un indice k : (x_k, y_k) , $k = 1, \dots, K$. toutefois il existe des variations de l'ensemble de données: a) certains composants des vecteurs x_k peuvent être manquant, b) les vecteurs x_k peuvent avoir des dimensions variables par construction, et c) les sorties y_k pourraient être spécifiées comme une plage de valeurs.

En ajustant une fonction d'agrégation aux données, nous distinguerons l'interpolation aux problèmes d'approximation. Dans le cas de l'interpolation, notre objectif est d'ajuster les valeurs de sortie spécifiées exactement. Par exemple, les paires $((0, 0, \dots, 0), 0)$ et $((1, 1, \dots, 1), 1)$ devraient toujours être interpolées. D'un autre côté, quand les données proviennent d'une expérience, il contiendra normalement quelques erreurs, et il est donc inutile d'interpoler les valeurs inexacts y_k . Dans ce cas, notre objectif est de rester proche des sorties souhaitées sans les faire correspondre. C'est le problème d'approximation.

Il y a bien sûr d'autres problèmes à prendre en compte lors du choix d'une fonction d'agrégation, tels que la simplicité, l'efficacité numérique, la facilité d'interprétation, etc.

[143]. Il n'y a pas de règles générales ici, et il appartient au développeur du système de faire un choix éclairé. Dans ce qui suit, nous nous concentrons sur les deux premiers critères: être cohérent avec les propriétés sémantiquement importantes de la procédure d'agrégation et ajuster les données souhaitées. Nous formalisons maintenant le problème de sélection.

Problème (Sélection d'une fonction d'agrégation). Prenons un certain nombre de propriétés mathématiques P_1, P_2, \dots et les données $D = \{(x_k, y_k)\}$ avec k allant de 1 à K . Choisissez une fonction d'agrégation f cohérente avec P_1, P_2, \dots , et satisfaisant $f(x_k) \approx y_k, k = 1, \dots, K$.

Les égalités approximatives peuvent bien sûr être satisfaites exactement, si les propriétés P_1, P_2, \dots permettent ceci. Nous considérerons également une variation du problème de sélection lorsque y_k est donné en intervalles, auquel cas nous avons besoin de $f(x_k) \in [y_{k\min}, y_{k\max}]$, ou même de satisfaire approximativement cette condition. La satisfaction des égalités approximatives $f(x_k) \approx y_k$ est habituellement traduite dans le problème de minimisation suivant.

$$\begin{aligned} & \textit{Minimiser} \|r\| \\ & \textit{sous réserve que } f \textit{ vérifie } P_1, P_2, \dots \end{aligned} \quad (58)$$

où $\|r\|$ est la norme des résidus, c'est-à-dire que $r \in \mathbb{R}^K$ est le vecteur de différences entre les valeurs prédites et observées $r_k = f(x_k) - y_k$. Il y a plusieurs façons de choisir la norme, et la plus populaire est la norme des moindres carrés.

$$\|r\|_2 = \left(\sum_{k=1}^K r_k^2 \right)^{1/2} \quad (59)$$

la norme d'écart le moins absolu

$$\|r\|_1 = \left(\sum_{k=1}^K |r_k| \right) \quad (60)$$

la norme Chebyshev

$$\|r\|_{\infty} = \max_{k=1, \dots, K} |r_k| \quad (61)$$

ou leurs analogues pondérés, comme

$$\|r\| = \left(\sum_{k=1}^K u_k r_k^2 \right)^{1/2} \quad (62)$$

où le poids $u_k \geq 0$ détermine l'importance relative du k-ième valeur y_k .

Exemple : Considérons que nous voulons choisir les poids d'une moyenne arithmétique pondérée compatible avec l'ensemble de données $\{(x_k, y_k)\}$ pour $k = 1$ à K , en utilisant l'approche des moindres carrés.

Nous minimisons la somme des carrés

$$\begin{aligned} & \text{Minimiser } \sum_{k=1}^K \left(\sum_{i=1}^n (w_i x_{ik} - y_k)^2 \right) \\ & \text{sous } \sum_{i=1}^n w_i = 1, \\ & w_1, \dots, w_n \geq 0 \end{aligned} \quad (63)$$

C'est un problème de programmation quadratique, qui est résolu par un certain nombre de méthodes standard.

Dans certaines études [144], il a été suggéré que pour les problèmes de prise de décision, la valeur numérique réelle du résultat $f(x_k)$ n'était pas aussi importante que le classement des résultats. Par exemple, si $y_k \leq y_l$, alors il devrait être $f(x_k) \leq f(x_l)$. En effet, les gens ne sont pas vraiment doués pour attribuer des scores numériques cohérents à leurs préférences, mais ils sont bons pour classer les alternatives. Ainsi, il est soutenu [144] qu'un choix approprié de la fonction d'agrégation devrait être cohérent avec le classement des sorties y_k plutôt qu'avec leurs valeurs numériques. L'utilisation des critères d'ajustement mentionnés ne préserve pas le classement des sorties, sauf si elles sont interpolées. La préservation du classement des sorties peut être faite en imposant les contraintes $f(x_k) \leq f(x_l)$ si $y_k \leq y_l$ pour toutes les paires k, l .

4.4 Conclusion

Nous avons décrit l'architecture détaillée de STROMa à travers la description des problèmes de décision, les critères et les alternatives. Le choix de l'opérateur qui est fondamental dans ces travaux de recherche a été discuté. L'utilisateur a la possibilité de sélectionner manuellement un opérateur d'agrégation pendant une procédure d'agrégation comme il peut se laisser proposer un opérateur d'agrégation par le système. Le type du problème (quantitatif ou qualitatif) et la catégorie du problème (choix, tri, rangement) influencent le choix de l'opérateur d'agrégation. Nous détaillons également l'algorithme des différents opérateurs mis en œuvre et enfin une étude sur comment choisir un opérateur d'agrégation face à un problème de décision.

Chapitre 5: Expérimentations

5.1 Méthodologie

Les systèmes de recommandation sont de plus en plus populaires à la fois commercialement et dans la communauté de la recherche, où de nombreuses approches ont été proposées pour fournir des recommandations. Afin de pouvoir comparer ces différents systèmes, il faut mettre en place diverses méthodes qui vont permettre de poser une base à partir de laquelle on va pouvoir comparer les différents systèmes de recommandation.

5.1.1 Outils disponibles

Afin d'effectuer des tests permettant d'obtenir un indice de performance correspondant à la capacité qu'a un système à produire de bonnes recommandations nous avons recherché des métriques. De bonnes recommandations permettent de satisfaire les utilisateurs en plaçant dans les premières places de listes qui leur sont présentées, des éléments qui leur plaisent.

Voici un aperçu des différentes méthodes qui existent afin d'obtenir un indice permettant de représenter la capacité qu'a un système à faire de bonnes recommandations.

Ces méthodes peuvent être rangées en 3 catégories. La première contient les méthodes pour mesurer la précision d'un système effectuant de la prédiction de note. La seconde contient les méthodes de mesure de prévision d'utilisation. Et enfin, la troisième contient les méthodes permettant d'évaluer les systèmes mettant en œuvre des tris des solutions/objets proposés. Ces méthodes sont tirées de [145]

5.1.1.1 Mesures de précision de la prédiction de note

Dans certaines applications, comme dans le service de location de film de Netflix (*netflix.com*), le système de recommandation tente de prédire la note qu'un utilisateur donnerait à un élément (par de 1 à 5 étoiles). Dans de tels cas, afin de mesurer la performance du système, il est nécessaire de mesurer la précision des prédictions.

L'écart quadratique moyen, ou Root Mean Squared Error (RMSE) est peut-être la métrique la plus couramment utilisée pour évaluer la précision des notes prédites. Le système génère des notes prédites NP_{uo} pour un ensemble de tests T pour des paires utilisateur-objet (u, o) pour lesquels les vraies notes NV_{uo} sont connues.

Expérimentations

En règle générale, les NV_{uo} sont connues bien qu'elles soient cachées lors d'une expérience hors ligne, ou parce qu'elles ont été obtenues grâce à une étude sur les utilisateurs ou une expérience en ligne.

L'écart quadratique moyen entre la note prédite et la note réelle est donné par la formule suivante :

$$RMSE = \sqrt{\frac{1}{|T|} * \sum_{(u,o) \in T} (NP_{uo} - NV_{uo})^2} \quad (64)$$

Une alternative très utilisée à RMSE est la mesure de l'erreur absolue moyenne, ou Mean Absolute Error (MAE).

$$MAE = \frac{1}{|T|} * \sum_{(u,o) \in T} |NP_{uo} - NV_{uo}| \quad (65)$$

Par rapport à MAE, RMSE pénalise de manière disproportionnée des erreurs importantes, de sorte que, étant donné un ensemble de tests avec quatre valeurs de RMSE d'objets cachés, la méthode RMSE produirait un meilleur résultat avec un système qui fait une erreur de 2 (RMSE=2) sur trois notes et 0 sur la quatrième, à celui qui fait une erreur de 3 (RMSE=3) sur une note et 0 sur tous les trois autres, tandis MAE ne préférerait que le second système.

5.1.1.2 Mesure de prévision d'utilisation

Dans de nombreux systèmes de recommandation, le système ne permet pas de prédire les préférences de l'utilisateur à propos d'éléments à lui proposer, tel que ce qui est fait lors de la prédiction de notes pour des films, mais essaie de recommander aux utilisateurs des éléments qu'ils pourraient utiliser (acheter), ou plus classiquement, qu'ils pourraient trouver intéressant (pertinent). Un exemple majeur de ces systèmes correspond aux moteurs de recherche.

Par exemple dans Netflix, lorsque des films sont ajoutés dans la liste personnelle des films de l'utilisateur (sa collection), le système propose un ensemble de films pouvant être intéressant, étant donné le film ajouté. Dans ce cas, nous ne nous intéressons pas à savoir si le système prévoit correctement les notations de ces films, mais plutôt de savoir si le système prévoit bien que l'utilisateur va ajouter ces films à sa collection (acheter le film).

Expérimentations

Afin d'évaluer les prévisions d'utilisation, nous possédons généralement un ensemble de données constituées des éléments que chaque utilisateur a utilisé (ajouté à sa collection par exemple), et qui sont donc considérés comme étant pertinents pour l'utilisateur. Ensuite, nous sélectionnons un utilisateur, nous masquons une partie de ses sélections, et nous utilisons le système de recommandation afin qu'il prévoie un ensemble d'éléments que l'utilisateur va utiliser. Nous obtenons alors quatre résultats possibles pour les articles recommandés et cachés. Ces résultats sont présentés dans le tableau suivant :

Tableau 10 : Classification des résultats possibles d'une recommandation d'un élément pour l'utilisateur

	Élément recommandé	Élément non recommandé
Élément pertinent	Vrai positif (vp)	Faux négatif (fn)
Élément non pertinent	Faux positif (fp)	Vrai négatif (vn)

Dans le cas où les tests se font de manière hors ligne (pas d'utilisation directe du système), puisque les données ne sont généralement pas recueillies par le système de recommandation au cours de l'évaluation, nous sommes obligés de supposer que les éléments non choisis ne l'auraient pas été, et ce même s'ils avaient été recommandés, c'est-à-dire qu'ils sont sans intérêt ou inutiles à l'utilisateur. Cette hypothèse peut être fautive, comme lorsque l'ensemble des articles non choisis contient quelques articles intéressants que l'utilisateur n'a pas choisis. Par exemple, un utilisateur peut ne pas avoir choisi un élément parce qu'il n'était pas au courant de son existence, mais après que le système de recommandation lui ait proposé cet élément, l'utilisateur peut décider de le sélectionner. Dans ce cas-là, le nombre de faux positifs est surestimé.

Nous pouvons ainsi compter le nombre d'éléments de chaque case du tableau et calculer les quantités suivantes :

$$precision = \frac{\text{nombre de } vp}{\text{nombre de } vp + \text{nombre de } nf} \quad (66)$$

La précision correspond au nombre d'éléments pertinents (vp) rapporté au nombre d'éléments total proposé par le système.

$$rappel = \frac{\text{nombre de } vp}{\text{nombre de } vp + \text{nombre de } fn} \quad (67)$$

Expérimentations

Le rappel est défini par le nombre d'éléments pertinents par rapport au nombre d'éléments pertinents total.

$$\text{Taux de vrai positif} = \frac{\text{nombre de fp}}{\text{nombre de fp} + \text{nombre de vn}} \quad (68)$$

Le taux de vrai positif est défini par le nombre de faux positifs par rapport au nombre d'éléments non pertinents

$$F - \text{mesure} = 2 * \frac{\text{precision} * \text{rappel}}{\text{precision} + \text{rappel}} \quad (69)$$

Cette mesure allie précision et rappel. C'est la moyenne harmonique de la précision et du rappel, la traditionnelle F-mesure.

Un système de recommandation parfait fournira des réponses dont la précision et le rappel sont égaux à 1. Dans ce cas-là, l'algorithme trouve la totalité des éléments pertinents (rappel), et il ne fait aucune erreur (précision). Dans la réalité, les systèmes sont plus ou moins précis, et plus ou moins pertinents. Il sera possible d'obtenir un système très précis (par exemple un score de précision de 0,99), mais peu performant (par exemple avec un rappel de 0.10, qui signifiera qu'il n'a trouvé que 10% des réponses possibles). Dans le même ordre d'idée, un algorithme dont le rappel est fort (par exemple 0.99 soit la quasi-totalité des éléments pertinents), mais la précision faible (par exemple 0.10) fournira en guise de réponse de nombreux éléments non pertinents en plus de ceux pertinents : il sera donc difficilement exploitable.

5.1.1.3 Mesure d'ordre (ranking measure)

Afin de mesurer l'ordre défini par un système de recommandation par rapport à un ordre de référence (bon ordre), plusieurs méthodes sont utilisées. Dans les cas où les notations (données explicites) sur les éléments, fournies par les utilisateurs, sont disponibles, nous pouvons classer les articles les mieux notés dans l'ordre décroissant des notes, avec des éléments liés. Il est important de noter que ces liaisons n'ont rien avoir avec les liens/dépendance/interactions entre des attributs. Dans notre cas, quand une paire d'objets est liée dans le classement établi (a obtenu la même note, ou le même score), cela signifie que

Expérimentations

l'utilisateur est en fait indifférent entre les deux objets. Ainsi, un système parfait ne doit pas classer un objet comme étant supérieur à l'autre.

Dans le cas où nous disposons seulement de données d'utilisation, il peut être approprié de construire un classement de référence où les éléments choisis par l'utilisateur sont classés au-dessus des éléments non choisis. Cependant, ceci n'est valable que si nous savons que l'utilisateur connaissait ces articles non choisis, de sorte que nous pouvons en déduire que l'utilisateur a en effet préféré les éléments choisis plutôt que les articles non choisis, et ce en toute connaissance de cause.

5.1.1.3.1 Normalized Distance-based Performance Measure (NDPM)

Dans certains systèmes il est nécessaire de définir un ordre total entre les éléments, et ce sans prendre en compte la présence ou non d'éléments liés. Dans ce genre de systèmes, l'évaluation ne doit pas pénaliser un système qui ordonne un élément par rapport à un autre alors que ceux-ci sont liés dans l'ordre de référence.

Dans ce cas, on utilise la méthode Normalized Distance-based Performance Measure (NDPM ; [146]), définie comme suit :

Avec $DI_{u\alpha}$ correspondant à l'index (place dans la liste) déterminé, et $RI_{u\alpha}$ correspondant à l'index de référence pour un objet et un utilisateur. Dans cette méthode on évalue toutes les paires possibles d'éléments constituant la liste. Cette évaluation des paires d'éléments consiste à déterminer si l'ordre défini par le système entre les éléments de la paire évaluée est le même dans l'ordre de référence.

Exemple :

Prenons une liste de référence (et donc un ordre de référence) : A, B, C. Et prenons une liste ordonnée générée par le système : A, C, B La méthode va évaluer les paires suivantes : AB, AC et BC.

Afin de définir NPDM il nous faut définir quelques valeurs :

$$C^+ = \sum_{ij} \text{sgn}(DI_{uoi} - DI_{uoj}) \text{sgn}(RI_{uoi} - RI_{uoj}) \quad (70)$$

$$C^- = \sum_{ij} \text{sgn}(DI_{uoi} - DI_{uoj}) \text{sgn}(RI_{uoj} - RI_{uoi}) \quad (71)$$

$$C^u = \sum_{ij} \text{sgn}^2(RI_{uoi} - RI_{uoj}) \quad (72)$$

$$C^s = \sum_{ij} \text{sgn}^2(DI_{uoi} - DI_{uoj}) \quad (73)$$

$$C^{uo} = C^u - (C^+ + C^-) \quad (74)$$

Donc C^u correspond au nombre de paires d'éléments pour lesquels l'ordre de référence définit un ordre entre les éléments, c'est-à-dire le nombre de paires d'éléments non liés (et idem pour C^s , mais avec l'ordre défini par le système). C^+ et C^- correspondent respectivement au nombre de paires d'éléments concordant et discordant. C'est-à-dire au nombre de paires d'éléments qui sont dans un ordre identique dans l'ordre de référence et l'ordre défini par le système. C^{uo} correspond au nombre de paires où le système de référence ne lie pas d'éléments, mais où le système le fait.

Si on reprend notre exemple :

La paire AB est concordante, la paire AC est concordante et la paire BC est non concordante.

NDPM est donc défini comme suit :

$$NDPM = \frac{C^- + 0.5 C^{uo}}{C^u} \quad (75)$$

Ainsi, la mesure NDPM donne un score parfait de 0 à un système qui permet de prédire correctement chaque relation de préférence par rapport à l'ordre de référence. Le pire score de 1 est attribué à un système qui contredit toutes les relations de préférence par rapport à l'ordre de référence. Ne pas prédire une relation de préférence est seulement pénalisé à moitié par rapport au fait de le contredire. Et également, on ne pénalise pas un ordre défini entre des éléments, qui sont liés dans l'ordre de référence (on ne connaît pas la préférence de l'utilisateur entre ces deux éléments).

5.1.1.3.2 Mesures de corrélation

Dans le cas où l'on possède un ordre total entre les éléments, et que lorsque deux éléments sont liés (cela signifie que l'utilisateur est en fait indifférent à la différence entre les deux

Expérimentations

objets.) l'on veut que le système ne donne pas d'ordre supérieur à l'un des deux éléments par rapport à l'autre, des mesures de corrélations (coefficient de corrélation) telles que celles de Spearman Rho ou Kendall Tau [147] [148] sont utilisées. On remarque que ces mesures tendent à être très similaires dans la pratique [149].

Le coefficient de Spearman Rho est calculé de la façon suivante (d_i étant la différence entre le rang de l'observation i et celui de sa valeur, et n est le nombre d'éléments) :

$$\rho = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n^3 - n} \quad (76)$$

Cependant la méthode Kendall Tau possède également une variante, Kendall Tau-b, qui effectue des ajustements afin de mieux prendre en compte les éléments liés, c'est pourquoi nous allons définir uniquement les méthodes Kendall Tau et Kendall Tau-b. On remarque que dans le cas où des éléments sont liés, chaque item lié devrait se voir assigner un rang moyen dans la liste, c'est-à-dire que si deux éléments sont liés à la 2ème et 3ème place, alors on leur assigne un rang de 2.5 [148].

Le coefficient de corrélation de Kendall Tau est défini comme suit :

$$\tau = \frac{C^+ - C^-}{\frac{1}{2}n(n-1)} \quad (77)$$

Il permet ainsi de mesurer à quel point l'ordre défini par le système est proche de l'ordre de référence. La valeur de ce coefficient est comprise entre -1 et 1. 1 correspond à un ordre défini par le système égal à l'ordre de référence. Tandis que -1 correspond à l'ordre inverse de l'ordre de référence. 0 correspond à la situation où l'ordre défini par le système n'est pas corrélé avec l'ordre de référence.

Afin de mieux prendre en compte les cas où les éléments sont liés entre eux, une variante dite Kendall Tau-b a été définie comme suit :

$$\tau_b = \frac{C^+ - C^-}{\frac{1}{2}\sqrt{(n(n-1) - U_x)(n(n-1) - U_y)}} \quad (78)$$

Expérimentations

Avec

$$U_x = \sum u_x(u_x - 1) \quad (79)$$

$$U_y = \sum u_y(u_y - 1) \quad (80)$$

Les symboles u_x et u_y correspondent respectivement au nombre d'apparitions de chaque valeur X et Y dans la liste d'éléments.

Pour donner quelques exemples, prenons une liste X définie par (A, B, C, D, E). Et une liste Y contenant les mêmes éléments, mais dans un ordre différent.

Tableau 11 : Exemple de valeur de coefficients de Kendall Tau-b

Coefficient de Kendall Tau-b par rapport à 2 listes	A, B, C, D, E
A, B, C, D, E	1
E, D, C, B, A	-1
A, B, C, E, D	0.8
A, B, E, C, D	0.6
A, C, D, E, B	0.4
A, C, E, D, B	0.2

Comme on peut le voir dans le Tableau 11, la valeur de 1 correspondant au classement parfait, et -1 au classement parfait inversé, et 0 correspondant à un classement ne se rapprochant pas de l'ordre parfait. On remarque que le fait d'avoir une valeur de 0.8 correspond à une situation presque parfaite (E a été déplacé d'un rang). De même, une valeur de 0.6 correspond à une situation très proche du parfait (E a été déplacé de 2 rangs). La valeur de 0.4 correspond à une situation qui commence à être moins correcte, mais qui tout de même offre un ordre de qualité acceptable (B a été déplacé de 3 rangs). La valeur de 0.2 correspond à une situation peu correcte, en effet l'ordre commence à ne plus être proche de la situation parfaite (B a été déplacé de 3 rangs et E de 1 rangs).

Ainsi on peut affirmer que des valeurs supérieures ou égales à 0.4 peuvent être considérées comme étant bonnes et celles supérieures à 0.6 comme étant très bonnes.

5.1.1.3.3 Méthode de K-cross fold

Pour effectuer des tests, on utilise fréquemment une méthode appelée K-cross fold. Cette méthode est utilisée dans le domaine de l'apprentissage pour effectuer des tests afin d'évaluer les capacités d'un algorithme d'apprentissage, et ce en réutilisant plusieurs fois la même base de test.

Voici son principe :

1. En fonction de k , on divise la base de données de test en k parties,
2. On sélectionne la première k partie, et on la met de côté, cette k partie sera utilisée en tant que base de données de validation, alors que les autres parties vont être utilisées en tant que base d'apprentissage,
3. On utilise dans le système, comme données en entrée, la base d'apprentissage que l'on vient de définir,
4. On teste ensuite le système sur la base de validation. Et grâce aux données obtenues, on va pouvoir utiliser diverses métriques pour évaluer le système.
5. On répète les opérations 2 à 4, en changeant à chaque fois la k partie utilisée, afin d'utiliser toutes les k parties en tant que base de validation.

Afin d'évaluer un système, cette méthode est souvent employée, et à l'issue des k validations, on calcule généralement la moyenne des résultats obtenus par les métriques employées.

5.1.2 Sélection des outils d'évaluation

Pour les expérimentations mises en place, il nous a fallu choisir quelles méthodes de mesures nous allions employer.

Le système que nous souhaitons tester n'utilise pas de système de prédiction de notes, donc les mesures définies à la section 5.1.1.1, ne peuvent être utilisées.

Le système que nous mettons en place a pour but de fournir une liste d'éléments à proposer à l'utilisateur qui soit dans un ordre qui corresponde à ses préférences. C'est pourquoi les méthodes définies dans la section 5.1.1.2, ne sont pas utilisables.

De plus, le système mis en place a pour but de construire un ordre total entre les différents éléments qui sont proposés à l'utilisateur. On remarque ainsi que les métriques

Expérimentations

définies à la section 5.1.1.3 correspondent tout à fait. Or dans le système mis en place, on ne définit pas d'ordre entre deux éléments liés, et dans notre système, il est fréquent d'avoir des éléments liés. C'est donc naturellement que lors des tests mis en place, on utilise comme mesure de performance, la mesure de coefficient de corrélation Spearman Rho. Cette mesure de performance reflétant à quel point les recommandations sont bonnes, c'est-à-dire à quel point l'ordre des éléments qui sont proposés à l'utilisateur sont dans le bon ordre.

5.2 Expérimentations

Afin de tester les performances de notre système, nous avons effectué plusieurs tests. Dans un premier temps nous avons comparé notre système avec celui de la plateforme MovieLens qui est un service de diffusion de films en ligne. Puis nous avons testé notre système avec de vrais utilisateurs, qui ont utilisé directement notre système. Ces différents tests ont permis de mettre en évidence l'efficacité du système mis en œuvre.

5.2.1 MovieLens

Nous utilisons un ensemble de données sur des films issus de la base MovieLens (accessible à l'adresse : « *movielens.org* »), qui est un service gratuit offert par le groupe de recherche GroupLens à l'Université du Minnesota. Cette base de données (BDD) est très utilisée dans le domaine de la recommandation [150]. Elle contient des données tirées des actions (donner une note à un film) de vrais utilisateurs. Cependant, ces utilisateurs n'ont pas utilisé directement notre système.

Des tests sont réalisés sur cette base de données de films enrichie au niveau de la quantité d'informations qu'elle contient. Ces tests ont pour but d'évaluer les performances du système mis en œuvre dans cette thèse.

5.2.1.1 Méthodologie

Nous voulions une base de données multicritère avec des notes de la part des utilisateurs et également que chaque utilisateur ait noté plusieurs objets.

C'est pourquoi nous avons utilisé la base de données issue de MovieLens. En effet, cette base de données correspond à ce que nous recherchons et de plus, elle est souvent utilisée dans le but d'étudier certains systèmes de recommandation. Les membres de

Expérimentations

MovieLens l'utilisent, et ce, afin d'apprendre à construire de meilleurs systèmes de recommandation. Cette base de données contient 10 millions de notes et 100 000 tags (informations supplémentaires données par les utilisateurs sous forme de mots-clés) pour 10 681 films et pour 71 567 utilisateurs. Les évaluations sont basées sur une échelle de nombre entier allant de 1 à 5.

Pour tester notre système par rapport aux données de MovieLens, Nous avons créé un problème de décision nommé film, puis nous avons considéré les 5 critères suivants pour ce problème de décision:

- Titre du film
- Auteurs
- Acteurs
- Genre
- Date de sortie

Un poids a été affecté à chaque critère en fonction de son degré d'importance. Ainsi, le titre s'est vu affecté un poids de 0.2, auteurs 0.1, acteurs 0.15, date de sortie 0.15 et genre 0.4. Ensuite, nous avons importé les données de MovieLens à travers de fichiers Excel. Ces données concernent la liste des films, la liste des utilisateurs, puis les notes attribuées par les utilisateurs. Nous avons sélectionné 30 utilisateurs dans le cadre de nos tests. Ces utilisateurs ont été sélectionnés de manière aléatoire. Après l'importation des notes, nous procédons à la désagrégation de ces notes afin de remplir la matrice de performance. Cette matrice de performance est utilisée par les différentes méthodes d'agrégation. Nous utilisons la formule suivante pour la désagrégation de notes :

$$S_{ij} = \frac{N_i * P_j}{\text{nombre de critères}} \quad (81)$$

Où S_{ij} représente la performance du critère j par rapport à l'alternative i . N_i représente la note attribuée par l'utilisateur à l'alternative (film) i . P_j est le poids du critère j .

5.2.1.2 Résultats

Tableau 12 : Résultats de l'utilisateur 1

Film	Ranking de l'utilisateur	Ranking de Méthode automatique (MOORA)	Ranking Méthode COPRAS	Ranking Méthode WASPAS
858-Godfather, The (1972)	1	1	1	1
1221-Godfather: Part II, The (1974)	1	1	1	1
1246-Dead Poets Society (1989)	1	1	1	1
2918-Ferris Bueller's Day Off (1986)	1	1	1	1
4878-Donnie Darko (2001)	1	1	1	1
5577-Igby Goes Down (2002)	1	1	1	1
59315-Iron Man (2008)	1	1	1	1
68358-Star Trek (2009)	1	1	1	1
69844-,Harry Potter and the Half-Blood Prince (2009)	1	1	1	1
73017-Sherlock Holmes (2009)	1	1	1	1
81834-Harry Potter and the Deathly Hallows: Part 1 (2010)	1	1	1	1
91542-Sherlock Holmes: A Game of Shadows (2011)	1	1	1	1
92439-Art of Getting By, The (2011)	1	1	1	1
96821-Perks of Being a Wallflower, The (2012)	1	1	1	1
112552-Whiplash (2014)	1	1	1	1
147-Basketball Diaries, The (1995)	16	16	16	16
2762-Sixth Sense, The (1999)	16	16	16	16
1968-Breakfast Club, The (1985)	18	18	18	18
2959-Fight Club (1999)	18	18	18	18
4226-Memento (2000)	18	18	18	18
33794-Batman Begins (2005)	18	18	18	18
58559-Dark Knight, The (2008)	18	18	18	18
99114-Django Unchained (2012)	18	18	18	18
54503-Superbad (2007)	24	24	24	24
91500-The Hunger Games (2012)	25	25	25	25
110-Streetcar Named Desire, A (1951)	26	26	26	26
98809-Hobbit: An Unexpected Journey, The (2012)	27	27	27	27

Le résultat obtenu par la méthode automatique, qui utilise la méthode MOORA, a un classement identique à celui du classement de l'utilisateur. Ce qui donne un coefficient de

Expérimentations

corrélation de Spearman Rho égal à 1. On retrouve ce résultat pour l'ensemble des méthodes d'agrégation implémentées. Nous avons conclu que la méthode de désagrégation utilisée est trop simpliste et ne prend pas en compte les films non évalués par l'utilisateur. Cela nous a poussés à chercher d'autres méthodes pour tester les performances du système mise en œuvre.

5.2.2 Choix d'un cuisinier

Nous voulons départager des cuisiniers selon leur aptitude à cuisiner trois mets. C'est un exemple proposé par Marichal & Roubens (2000) [151]. Soient les notations suivantes :

- X : ensemble des cuisiniers évalués selon leur aptitude à préparer trois mets : les cuisses de grenouilles (CG), le steak tartare (ST) et les coquilles Saint-Jacques (SJ).
- O : quatre cuisiniers A, B, C, D dont les évaluations (utilités), sur l'échelle $[0, 20]$, sont données ci-après :

Tableau 13 : Matrice d'utilité des 4 cuisiniers

Cuisiniers	CG	ST	SJ
A	18	15	19
B	15	18	19
C	15	18	11
D	18	15	11

Raisonnement du décideur :

- lorsqu'un cuisinier est reconnu pour sa préparation des coquilles Saint-Jacques, il est pensable qu'il prépare mieux les cuisses de grenouilles que le steak tartare ;
- à l'inverse, lorsqu'un cuisinier ne prépare pas bien les coquilles Saint-Jacques, il est pensable qu'il prépare mieux le steak tartare que les cuisses de grenouilles.

À partir de la matrice d'utilité, on peut voir que A et B préparent bien les coquilles Saint-Jacques (SJ). Dans ce cas, il est pensable qu'ils sachent mieux préparer les cuisses de grenouilles (CG). Il se trouve que A sait mieux préparer les cuisses de grenouilles que B d'où on peut déduire que A est préféré à B selon le décideur. C et D ne savent pas très bien préparer les coquilles Saint-Jacques, par conséquent, il est pensable qu'ils sachent mieux préparer le steak tartare que les cuisses de grenouilles. On peut constater que C sait

Expérimentations

mieux préparer le steak tartare que D, d'où C est préféré à D selon le décideur. Ainsi, on peut conclure que selon le décideur $A \geq B$ et $C \geq D$

En appliquant notre système sur cet exemple en utilisant la méthode automatique du choix de l'opérateur d'agrégation, nous avons le résultat suivant :

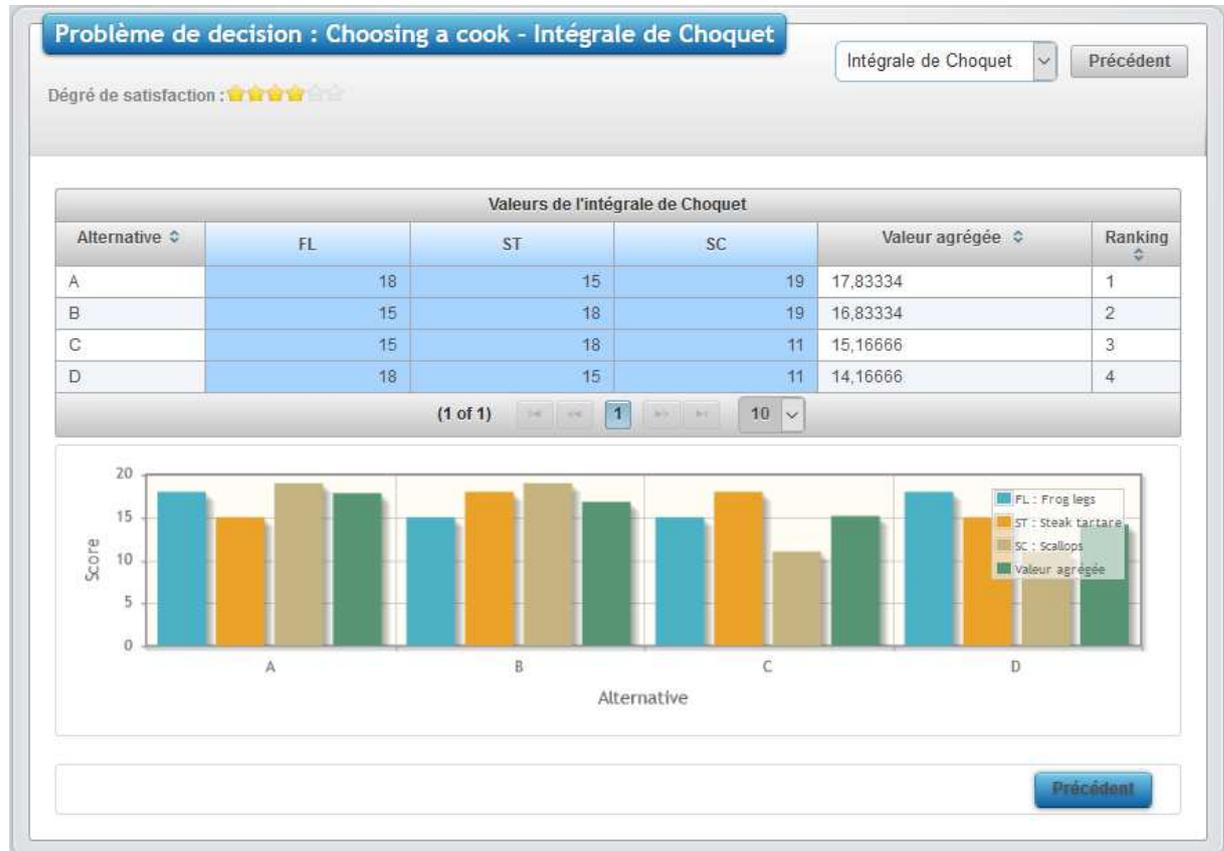


Figure 11 : Choix d'un cuisinier, méthode automatique

Nous pouvons voir que le système a utilisé la méthode de l'intégrale de Choquet dans la résolution de ce problème. Il est facile de constater que les préférences du décideur à savoir A préféré à B et C préféré à D ont été respectées. L'intégrale de Choquet a été sélectionnée car c'est une méthode quantitative et en plus, elle a le meilleur score parmi les autres opérateurs sur les problèmes de choix pour cet utilisateur. Ce score est calculé sur la base des degrés de satisfaction préalablement émis par l'utilisateur ou attribués par le système de manière implicite.

Voici un tableau comparatif entre le résultat de la méthode automatique et deux autres méthodes d'agrégation choisies au hasard :

Tableau 14 : choix du cuisinier, comparaison entre la méthode automatique et deux autres.

Cuisiniers	Ranking Utilisateur	Ranking Auto (Intégrale de Choquet)	TOPSIS	COPRAS
A	1	1	1	1
B	2	2	2	2
C	3	3	4	4
D	4	4	3	3

Le coefficient de corrélation de Spearman Rho est de 1 pour la méthode automatique, ce qui signifie un ordre parfait. Tandis qu'il est de 0.8 pour les méthodes TOPSIS et COPRAS. Ce score bien qu'il soit très élevé, est inférieur au score parfait de la méthode automatique à savoir l'intégrale de Choquet.

5.2.3 Sélection d'un robot industriel

Nous avons testé notre système sur l'exemple de sélection de robot industriel le plus approprié, proposé par Chakraborty et Zavadzka dans [152]. Les robots sont évalués sur 5 critères de 1 à 5. Parmi les cinq critères, seul C2 est un critère non bénéfique, c'est-à-dire un critère à minimiser et le reste de quatre critères sont des critères favorables, c'est-à-dire des critères à maximiser. Le tableau suivant donne la matrice de performance ainsi que des informations sur le poids des critères

Tableau 15 : Exemple du problème de sélection de robot

Weights	0.036	0.326	0.192	0.326	0.12
Alternative robots	C ₁	C ₂	C ₃	C ₄	C ₅
A ₁	60	0.4	2540	500	990
A ₂	6.35	0.15	1016	3000	1041
A ₃	6.8	0.1	1727.2	1500	1676
A ₄	10	0.2	1000	2000	965
A ₅	2.5	0.1	560	500	915
A ₆	4.5	0.08	1016	350	508
A ₇	3	0.1	1778	1000	920

Expérimentations

À noter que ces données ne sont pas normalisées. Elles seront normalisées au début de la procédure d'agrégation. En lançant la procédure d'agrégation sur cet exemple avec le mode automatique, nous obtenons les résultats suivants :

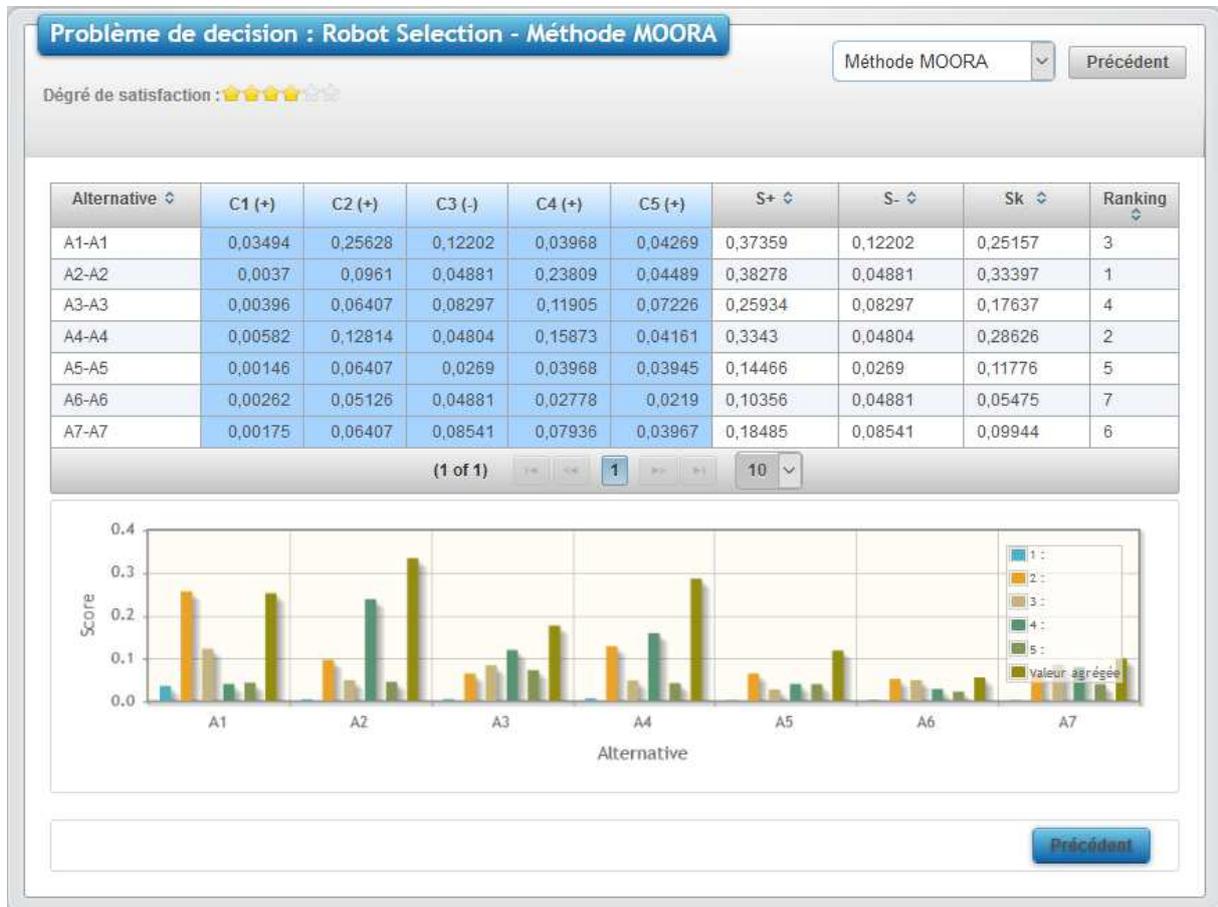


Figure 12 : Résultat de la sélection de robot

La méthode sélectionnée pour la résolution de ce problème a été la méthode MOORA avec un degré de satisfaction « Bon ». La méthode MOORA a été choisie car c'est un opérateur quantitatif et possède le meilleur score sur les problèmes de tri pour cet utilisateur. Après l'opération d'agrégation, l'utilisateur a la possibilité de renforcer ou d'affaiblir cet opérateur d'agrégation sur cette catégorie de problème de décision, selon son degré de satisfaction. Cela permet au système de continuer d'apprendre les préférences de l'utilisateur au fur et à mesure de l'utilisation du système. Par ailleurs, l'utilisateur n'est pas obligé de choisir la méthode automatique lors d'une procédure d'agrégation. En effet, il peut lui-même sélectionner un opérateur d'agrégation qui lui paraît compatible avec son problème de décision. Ce choix manuel d'un opérateur d'agrégation permet un renforcement automatique de cet opérateur vis-à-vis de cette catégorie de problème de décision.

Un tableau comparatif entre les différentes méthodes d'agrégation sur la sélection de robot industriel.

Tableau 16 : comparaison des différentes méthodes sur la sélection de robot

Alternative	Somme Pondérée	Intégrale de Choquet	MOORA	COPRAS	EDAS	WASPAS	TOPSIS
A1	1	1	3	2	1	3	3
A2	2	2	1	1	2	1	1
A3	4	3	4	4	4	4	4
A4	3	4	2	3	3	2	2
A5	6	6	5	5	6	5	5
A6	7	7	7	7	7	7	7
A7	5	5	6	6	5	6	6

Ce tableau donne un récapitulatif des résultats des différentes méthodes sur la sélection de robot. On peut noter une équivalence entre les résultats des méthodes MOORA, WASPAS et TOPSIS avec un coefficient de Spearman Rho égal à 1. Entre ces 3 méthodes et l'intégrale de Choquet, nous avons un coefficient de Spearman Rho égal à 0.78. Ce qui signifie un bon degré de similitude entre ces résultats. Entre les méthodes MOORA et COPRAS nous avons un coefficient de Spearman Rho égal à 0.96, ce qui est très proche à un ordre parfait.

5.3 Conclusion

Nous avons décrit les différentes méthodologies de test à savoir les mesures de précision de la prédiction de note, les mesures de prévision d'utilisation et enfin les mesures d'ordre. Nous avons choisi le coefficient de corrélation de Spearman Rho qui est une mesure d'ordre afin d'évaluer les performances de notre système. Nous avons effectué une série de tests. Le premier test a consisté à comparer les performances de notre système avec celui de MovieLens. Nous avons eu des résultats pratiquement parfaits à cause de la méthode de désagrégation assez simpliste. Nous avons alors testé notre système avec deux exemples classiques de la littérature qui ont produit des résultats prometteurs. Pour plus d'information sur les tests voir un cas complet dans les annexes.

Expérimentations

Conclusion générale

Il convient de rappeler les préoccupations qui étaient les nôtres au début de ces travaux de recherche. En effet, nous voulions mettre en place un système de recommandation mettant en œuvre plusieurs opérateurs d'agrégation et capable d'assister l'utilisateur par rapport au choix d'un opérateur face à un problème spécifique.

Afin d'atteindre ces objectifs, nous avons d'abord étudié les systèmes d'aide à la décision notamment les systèmes d'aide à la décision multicritères et les systèmes interactifs d'aide à la décision. Cela nous a permis de dégager pour qui cette aide était destinée et par qui elle provenait. Nous avons aussi rappelé des termes incontournables dans l'analyse multicritères tels que les critères et les actions. Le système de recommandation mis en œuvre exploite des caractéristiques des systèmes interactifs d'aide à la décision, à savoir l'utilisation du feedback de l'utilisateur afin d'affiner les recommandations.

Puis nous avons analysé différents opérateurs d'agrégation dans la littérature. A savoir la somme pondérée, l'intégrale de Choquet, l'intégrale de Sugeno, MOORA, COPRAS, EDAS, WASPAS et TOPSIS. Ces opérateurs ont été classés dans quatre grandes catégories à savoir des opérateurs de compromis, conjonctifs, disjonctifs et mixtes. Des propriétés mathématiques ont été dégagées pour ces différents opérateurs d'agrégation. Cela a permis d'avoir les points forts et les points faibles de chaque opérateur dans l'optique de le sélectionner sur d'éventuel problème de décision.

Le système de recommandation mis en place dispose en son sein de plusieurs opérateurs d'agrégation. Cela offre à l'utilisateur la possibilité de choisir une méthode d'agrégation adaptée à son problème de décision. Le système peut aussi intervenir au niveau du choix de l'opérateur d'agrégation. En effet, l'utilisateur peut se laisser proposer un opérateur d'agrégation par le système de manière automatique. Ce choix automatique est effectué en fonction des paramètres du problème de décision. A savoir si c'est un problème dont les évaluations sont quantitatives ou qualitatives. Il y a aussi la catégorie du problème qui concerne l'objectif visé du problème. Cela peut être un problème de choix, un problème de tri ou un problème de rangement. Enfin un système de notation est mis en place permettant de récolter la satisfaction du décideur vis-à-vis d'un opérateur d'agrégation face à un problème de décision. C'est sur la base des notes attribuées par l'utilisateur explicitement, ou

Conclusion générale

implicitement par le système, que l'opérateur d'agrégation ayant récolté le plus grand degré de satisfaction dans la catégorie est proposé.

Après avoir présenté des outils d'expérimentation, nous avons effectué plusieurs tests afin de valider les performances de notre système. Une première comparaison a été faite sur la base de données de MovieLens. Cela a semblé donner des résultats presque parfait à cause de la méthode de désagrégation simpliste utilisée. Nous avons testé le système sur deux autres exemples qui ont produit des résultats encourageants.

Les perspectives à notre travail sont nombreuses, voici quelques-unes d'entre elles :

- La première perspective concerne l'intégration d'autres opérateurs d'agrégation dans le système. En effet nous avons implémenté moins de dix opérateurs d'agrégation alors qu'il existe des dizaines voire une centaine d'opérateurs d'agrégation dans la littérature.
- La deuxième, il s'agit de créer de nouveaux opérateurs d'agrégation lorsque cela s'avère nécessaire.
- L'amélioration de la méthode de sélection automatique de l'opérateur d'agrégation. En effet, actuellement nous utilisons quelques paramètres et un système de notation afin de sélectionner l'opérateur le plus approprié. Nous voudrions pouvoir dans le futur, exploiter au maximum les propriétés des opérateurs d'agrégation dans le choix automatique. Nous voudrions traiter le choix de l'opérateur d'agrégation comme un problème d'optimisation comme décrit dans le chapitre 4 section 4.3. En effet la résolution de ce problème d'optimisation fait partie des perspectives de ce travail.
- Elargir le cadre de cette recherche pour y inclure des procédures d'agrégation de la programmation mathématique multiobjectif,
- Inclure des procédures pour la décision participative et/ou de négociation

Références

- [1] B. Roy, “Decision Science or Decision-Aid Science?,” *European Journal of Operational Research*, vol. 66, pp. 184-203, 1993.
- [2] A. Schärliig, *Décider sur plusieurs critères.*, Lausanne: Presses polytechniques romandes, 1985.
- [3] V. M. Ozernoy, “Choosing the ‘Best’ Multiple Criteria Decision-Making Method,” *INFOR*, vol. 30, no. 2, pp. 159-171, 1992.
- [4] A. TSOUKIÀS, *De la théorie de la décision à l’aide à la décision.*
- [5] K. BOWEN, “Sixty years of operational research,” *European Journal of Operational Research*, vol. 153, p. 618–623, 2004.
- [6] M. W. KIRBY, *A history of Operational Research in Britain*, London: World Scientific, 2002.
- [7] L. EULER, “Solutio problematis ad geometriam situs pertinentis,” *Opera Omnia*, vol. 7, p. 128–140, 1736.
- [8] J. C. BORDA, “Mémoire sur les élections au scrutin. Comptes rendus de l’Académie des sciences, traduit par Alfred de Grazia comme Mathematical Derivation of a election system,” *Isis*, vol. 44, pp. 42-51, 1781.
- [9] MARQUIS DE CONDORCET, *Essai sur l’application de l’analyse à la probabilité des décisions rendues à la pluralité des voix.* Imprimerie Royale, Paris, 1785.
- [10] V. PARETO, *Manuale di Economia Politica.* Piccola Biblioteca Scientifica. Translated into English by Ann S. Schwier (1971), *Manual of Political Economy*, MacMillan, London., Milan, 1906.
- [11] V. PARETO and H. FAYOL, *General and Industrial Management*, New York: Pitman and Sons, 1949.
- [12] F. W. TAYLOR, *The principles of Scientific Management*, New York: Harper and Row, 1911.
- [13] A. TURING, “On computable numbers, with an application to the entscheidungsproblem,” *Proceedings of the London Mathematical Society*, vol. 42, p. 230–265, 1937.
- [14] G. B. DANTZIG, “Programming in a linear structure.,” USAF, Washington D.C, 1948.
- [15] L. V. KANTOROVICH, “Mathematical methods in the organisation and planning of production.,” *Publication House of the Leningrad State University, Leningrad, Translated into english in: Management Science*, vol. 6, no. 960, p. 366–422, 1939.
- [16] J. F. NASH, “The bargaining problem.,” *Econometrica*, vol. 18, p. 155–162, 1950.
- [17] J. F. NASH, “Non cooperative games.,” *Annals of Mathematics*, vol. 54, p. 286–295, 1951.
- [18] J. VON NEUMANN and O. MORGENSTERN., *Theory of games and economic behavior.*, Princeton: Princeton University Press, 1944.
- [19] A. TURING, “Computing machinery and intelligence.,” *Mind*, vol. 49, p. 433–460, 1950.
- [20] P. M. BLACKETT, “Operational research,” *Operational Research Quarterly*, vol. 1,

- pp. 3-6, 1950.
- [21] G. B. DANTZIG, "Application of the simplex method to a transportation problem," *Activity Analysis of Production and Allocation (T.C. Koopmans, ed.) Wiley*, pp. 359-373, 1951.
- [22] N. WIENER, *Cybernetics*, Cambridge: MIT Press, 1948.
- [23] W. R. ASHBY, *An introduction to cybernetics*, London: Chapman & Hall, 1956.
- [24] K. J. ARROW, *Social choice and individual values*, New York: J.Wiley, 1951 2nd edition, 1963..
- [25] D. BOUYSSOU, "Democracy and efficiency - A note on 'Arrow's theorem is not a surprising result' .," *European Journal of Operational Research*, vol. 58, p. 427-430, 1992.
- [26] P. VINCKE, "Aggregation of preferences: a review," *European Journal of Operational Research*, vol. 9, pp. 17-22, 1982.
- [27] J. S. KELLY, *Arrow Impossibility Theorems.*, New York: Academic Press, 1978.
- [28] J. S. KELLY, "Social choice bibliography," *Social Choice and Welfare*, vol. 8, pp. 97-169, 1991.
- [29] H. NURMI, "Comparing voting systems. D. Reidel, Dordrecht," 1987.
- [30] H. NURMI, *Voting paradoxes and how to deal with them?*, Berlin: Springer Verlag, 1999.
- [31] A. TAYLOR, *Mathematics and Politics: Strategy, Voting, Power, and Proof.*, Berlin: Springer Verlag, 1995.
- [32] M. ALLAIS, "Le comportement de l'homme rationnel devant le risque : Critique des postulats et axiomes de l'école américaine. *Econometrica*," *ECONOMETRICA*, vol. 21, no. 4, pp. 503-546, 1953.
- [33] C. H. COOMBS, "On the use of inconsistency in preferences in psychological measurement.," *Journal of Experimental Psychology*, vol. 55, pp. 1-7, 1958.
- [34] K. O. MAY, "Intransitivity, utility and the aggregation of preference patterns.," *Econometrica*, vol. 22, pp. 1-13, 1954.
- [35] H. A. SIMON, *Administrative behaviour: a study of Decision Making Processes in Administrative Organizations.*, New York: Mac Millan, 1947.
- [36] H. A. SIMON, "A behavioral model of rational choice.," *Quarterly Journal of economics*, vol. 69, pp. 99-118, 1954.
- [37] H. A. SIMON, "Rational choice and the structure of the environment.," *Psychological Review*, vol. 63, pp. 129-138, 1956.
- [38] H. A. SIMON, "A behavioural model of rational choice. In H.A. SIMON, Ed., *Models of man: social and rational; mathematical essays on rational human behavior in a social setting*, pages 241-260. J.Wiley, New York, 1957."
- [39] J. MOSCAROLA, "Organizational decision processes and ORASA intervention. In R. TOMLINSON AND I. KISS, Eds, *Rethinking the process of operational research and systems analysis*," *Pergamon Press, Oxford*, pp. 169-186, 1984.
- [40] J. L. LE MOIGNE, *La Théorie du système général: Théorie de la modélisation.*, Paris: Presses Universitaires de France, 1977.
- [41] H. A. SIMON, *The science of the artificial.*, Cambridge: MIT Press, 1969.
- [42] P. VINCKE, *Multicriteria Decision-Aid.*, New York: J. Wiley, 1992.
- [43] C. BERGE, *Théorie des graphes et ses applications.*, Paris: Collection Univesitaire des

- Mathématiques, Dunod, 1958.
- [44] A. CHARNES and W. W. COOPER, Management models and industrial applications of linear programming., New York,: J. Wiley, 1961.
- [45] C. W. CHURCHMAN, R. L. ACKOFF and E. L. ARNOFF, Introduction to Operations Research., New-York: J. Wiley, 1957.
- [46] G. B. DANTZIG, Linear Programming and Extensions., Princeton: Princeton University Press, 1963.
- [47] R. FAURE, Éléments de la recherche opérationnelle., Paris: Gauthier-Villars, 1968.
- [48] S. I. GASS, Linear Programming: Methods and Applications., New York: Mc Graw Hill, 1958.
- [49] R. L. ACKOFF, “Some unsolved problems in problem solving.,” *Operational Research Quarterly* , vol. 13, pp. 1-11, 1962.
- [50] R. M. ADELSON and J. M. NORMAN, “Operational research and decision making.,” *Operational Research Quarterly*, vol. 20, pp. 399-413, 1969.
- [51] L. A. ZADEH, “Fuzzy sets.,” *Information Control*, vol. 8, p. 338–353, 1965.
- [52] W. EDWARDS, “The theory of decision making. 41:380–417, 1954.,” *Psychological Bulletin*, vol. 41, pp. 380-417, 1954.
- [53] L. FESTINGER, Conflict, decision and dissonance., Stanford: Stanford University Press, 1964.
- [54] W. LICHTENSTEIN and P. SLOVIC, “Reversals of preferences between bids and choices gambling decisions.,” *Journal of Experimental Psychology*, vol. 89, pp. 46-55, 1971.
- [55] W. LICHTENSTEIN, P. SLOVIC and D. ZINK , “Effect of instruction in expected value on optimality of gambling decisions.,” *Journal of Experimental Psychology*, vol. 79, pp. 236-240, 1969.
- [56] A. TVERSKY, “Additivity, utility and subjective probability.,” *Journal of Mathematical Psychology*, vol. 4, pp. 175-201, 1967.
- [57] A. TVERSKY, “Intransitivity of preferences.,” *Psychological Review*, vol. 76, pp. 31-48, 1969.
- [58] A. TVERSKY, “Features of similarity.,” *Psychological Review*, vol. 84, pp. 327-352, 1977.
- [59] B. DUSHNIK and E. W. MILLER, “Partially ordered sets.,” *American Journal of Mathematics*, vol. 63, pp. 600-610, 1941.
- [60] R. D. LUCE, “Semiorders and a theory of utility discrimination.,” *Econometrica*, vol. 24, pp. 178-191, 1956.
- [61] D. SCOTT and P. SUPPES, “Foundational aspects of theories of measurement.,” *Journal of Symbolic Logic*, vol. 23, pp. 113-128, 1958.
- [62] P. C. FISHBURN , Utility Theory for Decision Making., New York: Wiley, 1970.
- [63] D. H. KRANTZ, R. D. LUCE, P. SUPPES and A. TVERSKY, Foundations of measurement, volume 1: Additive and polynomial representations., New York: Academic Press, 1971.
- [64] P. C. FISHBURN, Interval Orders and Interval Graphs., New York: J. Wiley, 1985.
- [65] M. PIRLOT and P. VINCKE, Semi Orders., Dordrecht: Kluwer Academic, 1997.
- [66] F. S. ROBERTS, Measurement theory, with applications to Decision Making, Utility and the Social Sciences., Boston: Addison-Wesley, 1979.

- [67] M. ROUBENS and P. VINCKE, Preference Modeling., Berlin: LNEMS 250, Springer Verlag, 1985.
- [68] M. ÖZTÜRK, A. TSOUKIÀS and P. VINCKE, "Preference modelling. In M. EHRGOTT, S. GRECO AND J. FIGUEIRA, Eds, State of the Art in Multiple Criteria Decision Analysis, Berlin," *Springer Verlag*, pp. 27-72, 2005.
- [69] G. BATESON, Steps to an Ecology of Mind., Group, S. Francisco: Chandler Publ., 1972.
- [70] E. GOFFMAN, Asylums; essays on the social situation of mental patients and other inmates., Chicago: Alding Publishing Co, 1968.
- [71] P. WATZLAWICK, J. H. BEAVIN and D. D. JACKSON, Pragmatics of Human Communication., New York: W.W. Norton, 1967.
- [72] M. Abdolmohammadi, C. Usoff and B. College, "A longitudinal study of applicable decision aids for detailed tasks in a financial audit.," *Intelligent Systems in Accounting, Finance Management*, p. 139–154, 2001.
- [73] H. A. Simon, "Making management decisions : the role of intuition and emotion.," 2001.
- [74] A. ZIELKE, H. SITTER, T. RAMPP, T. BOHRER and M. ROTHMUND, "Clinical decision-making, ultrasonography, and scores for evaluation of suspected acute appendicitis.," *World Journal of Surgery*, vol. 5, no. 25, p. 578–584, 2001.
- [75] D. POOLE, "Decision-theoretic defaults. In Proceedings of the Ninth Biennial Conference of the Canadian Society for Computational Studies of Intelligence," *Morgan Kaufmann*, p. 190–197, 1992.
- [76] P. JOURNEE, P. PERNY and D. VANDERPOOTEN, "A multicriteria methodology for the verification of arms control agreements in Europe," *Foundations of Computing and Decision Sciences*, vol. 23, no. 2, p. 64–85, 1998.
- [77] P. WATZLAWICK, The situation is hopeless, but not serious: (the pursuit of unhappiness), New York: Norton, 1983.
- [78] A. CAPRARA, P. TOTH, D. VIGO and M. FISCHETTI, "Modeling and solving the crew rostering problem," *Operations Research*, vol. 46, p. 820–830, 1998.
- [79] A. OSTANELLO and A. TSOUKIÀS, "An explicative model of 'public' interorganizational interactions," *European Journal of Operational Research*, vol. 70, p. 67–82, 1993.
- [80] H. A. SIMON, "The new science of management decision.," 1977.
- [81] B. ROY, Méthodologie Multicritère d'Aide à la décision, Paris: ECONOMICA, 1985.
- [82] . L. Y. MAYSTRE, J. PICTET and J. SIMOS, Méthodes multicritères ELECTRE., Lausanne: Presses polytechniques et universitaires romandes, 1994, p. 323.
- [83] R. L. KEENEY and H. RAIFFA, Decisions with Multiple Objects., New York: John Wiley and Sons, 1976.
- [84] T. SAATY, Décider face à la complexité., Paris: Entreprise moderne d'édition., 1984, p. 231.
- [85] K. FRÄMLING, "Modélisation et apprentissage des préférences par réseaux de neurones pour l'aide à la décision multicritère, Thèse pour L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE LYON," 1996.
- [86] B. ROY, "Classement et choix en présence de points de vue multiples, la méthode ELECTRE.," *Revue informatique et recherche opérationnelle*, vol. 2, no. 8, pp. 57-75, 1968.

- [87] B. ROY, "ELECTRE III : un algorithme de classements fondé sur une représentation floue de préférences en présence de critères multiples.," *Cahiers du CERO*, vol. 20, no. 1, pp. 3-24, 1978.
- [88] J. P. BRANS , B. MARESCHAL and P. VINCKE , " PROMETHEE : a New Family of Outranking Methods in Multicriteria Analysis. In : Operational Research," *E dited by J.-P. Brans. Amsterdam :Elsevier Science Publishers B.V.*, pp. 408-421, 1984.
- [89] J. PAELINCK, "Qualiflex, a flexible multiple criteria method.," *Economic letters*, no. 3, pp. 193-197, 1978.
- [90] H. RAIFFA , *Decision Analysis.*, Massachusetts: Dearing, 1968.
- [91] H. RAIFFA, *Analyse de la décision.*, Paris: Dunod, 1973, p. 328 p.
- [92] R. SCHLAIFER , *Probability and Statistics for Business Decisions.*, New York: McGraw-Hill, 1959, p. 732 p.
- [93] R. SCHLAIFER , *Analysis of Decisions Under Uncertainty.*, New York: McGraw-Hill, 1969, p. 729 p.
- [94] P. VINCKE , *Vrai, quasi, pseudo et precritères dans un ensemble fini, propriétés et algorithmes.*, Paris: Technical report, Cahiers du LAMSADE, N° 27, Université Paris Dauphine, 1980.
- [95] B. ROY , *Méthodologie multicritère d'aide à la décision. Traduction anglaise : Multicriteria Methodology for Decision Aiding*, Dordrecht: Economica, 1985.
- [96] B. ROY and P. VINCKE, "Pseudo-orders : definition, properties and numerical representation.," *Mathematical Social Sciences 14*, pp. 263-274, 1987.
- [97] P. VINCKE, *L'aide multicritère à la décision*, Bruxelles: Éditions de l'Université de Bruxelles, 1989, p. 179 pages.
- [98] C. DONG, "Développement d'outils d'aide à la décision dans des environnements multicritères, dynamiques et incertains : Application à des problèmes de planification régionale. Thèse Doct. Ing," Louvain-la-Neuve, Belgique, 1992.
- [99] H. SIMON, "A behavioural model of rational choice," *Quarterly Journal of Economics*, vol. 69, pp. 99-118, 1955.
- [100] H. SIMON, *Administration et Processus de décision.*, Paris: Economica, coll Gestion, 1983.
- [101] D. HAMDADOU, K. LABED and A. BENYETTOU, "Un Système Interactif Multicritère d'Aide à la Décision en Aménagement du Territoire : Approche du Tri, Intégrale de Choquet et SIG", *SETIT 2007 4th International Conference : Sciences of Electronic, Technologies of Information, and Telecommunications*, 2007.
- [102] S. FOMBA, P. ZARATE, D. M. KILGOUR , G. CAMILLERI, J. KONATE and F. TANGARA, "A Recommender System Based on Multi-Criteria Aggregation," *International Journal of Decision Support System Technology (IJDSST)*, vol. 9, no. 4, pp. 1-15, 2017.
- [103] A. MARTIN, "Thèse: Evolution de profils Multi-Attributs, par Apprentissage Automatique et Adaptatif dans un système de Recommandation pour l'aide à la Décision," Université Toulouse 3 Paul Sabatier , Toulouse, 2012.
- [104] W. EDWARDS , "Social utilities," *Engeneering Economist, Summer Symposium Series*, vol. 6, pp. 119-129, 1971.
- [105] H. RAIFFA , "Preferences for multi-attributed alternatives, Rapport technique n°RM-58-68-DOT/RC, The Rand Corporation, Santa Monica, Californie," 1969.
- [106] D. BOUYSSOU, D. DUBOIS , M. PIRLOT and H. PRADE, *Concept et méthodes pour*

l'aide à la décision 3 - analyse multicritère, Paris: Lavoisier, 2006.

- [107] D. KRANTZ, R. LUCE, P. SUPPES and A. TVERSKY, Foundations of measurement, volume 1 : Additive and polynomial representations, Academic Press, 1971.
- [108] D. BOUYSSOU and M. PIRLOT, "Additive difference' models without additivity and subtractivity.," *Journal of Mathematical Psychology*, vol. 48, no. 3, pp. 263-291, 2004.
- [109] D. BOUYSSOU and M. PIRLOT, "Preferences for multi-attributed alternatives : Traces, dominance, and numerical representations," *J. of Mathematical Psychology*, vol. 48, pp. 167-185, 2004.
- [110] A. CORNUÉJOLS and L. MICLET, Apprentissage artificiel : concepts et algorithmes., Editions Eyrolles., 2011.
- [111] A. G. BARTO, Reinforcement learning : An introduction., MIT press., 1998.
- [112] R. YAGER, "On ordered weighted averaging aggregation operators in multicriteria decisionmaking," *IEEE Trans. Systems, Man & Cybern*, vol. 18, pp. 183-190, 1988.
- [113] M. SUGENO, "Theory of fuzzy integrals and its applications. Thèse doctorale," Tokyo Institute of technology, 1974.
- [114] G. CHOQUET, "Theory of capacities," *In Annales de l'Institut Fourier*, vol. 5, pp. 131-295, 1953.
- [115] L. S. SHAPLEY, A value for n-person games. Rapport technique, DTIC Document., 1952.
- [116] T. e. S. S. MUROFUSHI, Techniques for reading fuzzy measures (iii) : interaction index. In 9th Fuzzy System Symposium, 1993.
- [117] M. GRABISCH, Alternative representations of discrete fuzzy measures, Int. J. of Uncertainty, Fuzziness, and Knowledge Based Systems.
- [118] W. Brauers and . E. Zavadskas, "The MOORA method and its application to privatization in a transition economy.," *Control Cybern*, vol. 35, no. 2, p. 445-469, 2006.
- [119] X. Luo and N. Jennings, "A spectrum of compromise aggregation operators for multiattribute decision making.," *Artificial Intelligence*, vol. 171, no. (2-3), pp. 161-184, 2007.
- [120] C. Alsina, M. Frank and B. Schweizer, "Associative functions: triangular norms and copulas.," *World Scientific*, 2006.
- [121] M. PAZZANI and D. BILLSUS , "« Learning and Revising User Profiles : The Identification of Interesting Web Sites »,," *Machine Learning*, vol. 27, pp. 313-331, 1997.
- [122] Y. ZHANG , J. CALLAN and T. MINKA , "Novelty and Redundancy Detection in Adaptive Filtering, ? Proc.," *25th Ann. Int'l ACM SIGIR Conf.*, pp. 81-88, 2002.
- [123] T. Sørensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons.," *Kongelige Danske Videnskabernes Selskab*, vol. 5, no. 4, p. 1-34, 1948.
- [124] J. I. Rocchio, "Relevance feedback in information retrieval. In The SMART Retrieval System," *Prentice-Hall*, pp. 313-323, 1971.
- [125] D. BILLSUS and M. PAZZANI, "« Learning Collaborative Information Filters » ICML '98 : Proceedings of the Fifteenth International Conference on Machine Learning, San Francisco, CA.," *USA : Morgan Kaufmann Publishers Inc.*, pp. 46-54, 1998.
- [126] J. S. BREESE , D. HECKERMAN and C. KADIE, "« Empirical Analysis of Predictive

- Algorithms for Collaborative Filtering, » Proc.,” *14th Conf. Uncertainty in Artificial Intelligence*, pp. 43-52, 1998.
- [127] P. Chen and P. Popovich, *Correlation :Parametric and Nonparametric Measures*, Sage University Papers Series on Quantitative Applications in the Social Sciences, 2002.
- [128] G. U. Yule and M. G. Kendall, *An Introduction to the Theory of Statistics (14th ed.)*, Charles Griffin & Co. , 1968.
- [129] C. BASU , H. HIRSH and W. COHEN , “Recommendation as classification : using social and content-based information in recommendation,” *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, American Association for Artificial Intelligence, pp. 714-720, 1998.
- [130] A. I. SCHEIN, A. POPESCU, L. H. UNGAR and D. M. PENNOCK, “Methods and Metrics for Cold-Start Recommendations,” *Proc. 25th Ann. Int’l ACM SIGIR Conf.*, , 2002.
- [131] G. LINDEN, B. SMITH and J. YORK, “Amazon.com Recommendations : Item-to-Item Collaborative Filtering,” *IEEE Internet Computing*, vol. 7, pp. 76-80, 2003.
- [132] J. B. SCHAFER , D. FRANKOWSKI and J. HERLOCKER, “Collaborative Filtering Recommender Systems,” *The Adaptive Web : Methods and Strategies of Web Personalization*, pp. 291-324, 2007.
- [133] M. Roubens and P. Vincke, *Preference Modelling*, Berlin: Springer-Verlag, 1985.
- [134] P. Vincke, *L’aide multicritère à la décision.*, Éditions de l’Université de Bruxelles., 1989.
- [135] B. Roy and D. Bouyssou, *Aide Multicritère à la décision: Méthodes et Cas.*, Paris: Economica, 1993.
- [136] I. K. P. M. Michel Grabisch, *Non-Additive Measure and Integral Manipulation Functions*, 2015.
- [137] T. Mori and T. Murofushi, “An analysis of evaluation model using fuzzy measure and the Choquet integral,” in *‘5th Fuzzy System Symposium, Kobe, Japan, In Japanese.*, p. pp. 207–212, 1989.
- [138] M. Grabisch, “A new algorithm for identifying fuzzy measures and its application to pattern recognition,” in *‘Int. 4th IEEE Conf. on Fuzzy Systems’*, Yokohama, Japan., p. 145–150., 1995.
- [139] J. L. Marichal and M. Roubens, “Determination of weights of interacting criteria from a reference set,” *European Journal of Operational Research* 124, pp. 641-650, 2000.
- [140] P. Meyer and M. Roubens, “Choice, ranking and sorting in fuzzy Multiple Criteria Decision Aid, in J. Figueira, S. Greco & M. Ehrgott, eds, ‘Multiple Criteria Decision Analysis: State of the Art Surveys’,” *Springer*, pp. 471-506, 2005.
- [141] I. Kojadinovic, “Minimum variance capacity identification,” *European Journal of Operational Research* p. in press., 2006.
- [142] I. Kojadinovic, “Quadratic distances for capacity and bi-capacity approximation and identification,” *Quarterly Journal of Operations Research* p. in press., 2006.
- [143] H. J. Zimmermann, *Fuzzy Set Theory - and its Applications.*, Boston: Kluwer, 1996.
- [144] U. Kaymak and H. R. van Nauta Lemke, “Selecting an aggregation operator for fuzzy decision making,” in *3rd IEEE Intl. Conf. on Fuzzy Systems*, vol. 2, p. 1418–1422, 1994.
- [145] G. SHANI and A. GUNAWARDANA, “Evaluating Recommendation Systems, Recommender Systems Handbook,” *Springer*, pp. 257-297, 2011.

- [146] Y. Y. YAO, “Measuring retrieval effectiveness based on user preference of documents.,” *J. Amer. Soc. Inf. Sys.*, vol. 46, no. 2, pp. 133-145, 1995.
- [147] M. G. KENDALL, “A new measure of rank correlation.,” *Biometrika*, vol. 30, no. 1-2, pp. 81-93, 1938.
- [148] M. G. KENDALL , “The treatment of ties in ranking problems.,” *Biometrika*, vol. 33, no. 3, pp. 239-251, 1945.
- [149] G. A. FREDRICKS and R. B. NELSEN, “On the relationship between spearman’s rho and kendall’s tau for pairs of continuous random variables.,” *Journal of Statistical Planning and Inference*, vol. 137, no. 7, pp. 2143-2150, 2007.
- [150] “MILLER B.-N., ALBERT I., LAM S.-K., KONSTAN J.-A., RIEDL J., MovieLens Unplugged : Experiences with an Occasionally Connected Recommender System. In Proceedings of ACM 2003 International Conference on Intelligent User Interfaces (IUI’03) (Accepted Poster)”.
- [151] J. L. Marichal and M. Roubens, “Determination of weights of interacting criteria from a reference set,” *European Journal of Operational Research*, vol. 124, pp. 641-650, 2000.
- [152] S. Chakraborty and E. K. Zavadskas, “Applications of WASPAS method in manufacturing decision making.,” *Informatica*, vol. 25, no. 1, pp. 1-20, 2014.

Annexes

Exemple complet : Sélection du fournisseur logistique

Nous détaillons l'ensemble des étapes de ce problème de décision à travers des captures d'écran. C'est un problème de sélection des fournisseurs logistiques. Le problème de décision contient cinq critères et sept alternatives. Juste C3 est un critère à minimiser et le reste des quatre critères sont des critères à maximiser.

La création du problème se fait en suivant les étapes Description, Performances, Préférences et Validation. Une fois le problème de décision créé on peut procéder à l'agrégation des performances afin d'établir un classement. Nous utilisons tout d'abord la méthode automatique par rapport au choix de l'opérateur sur ce problème de décision. Puis nous utilisons la méthode manuelle en sélectionnant tour à tour les autres opérateurs d'agrégation. Cela permet de faire une comparaison entre la méthode proposée automatiquement et les autres méthodes d'agrégation.

The screenshot shows a software interface for creating a decision problem. The title bar reads "Problème de décision : Logistic provider selection". The interface is divided into four tabs: "Description" (1), "Performances" (2), "Préférences" (3), and "Validation" (4). The "Description" tab is active, showing the "Description des critères" section. This section includes a "Description du problème" area with fields for "Code" (LPS), "Libellé" (Logistic provider selection), and "Nombre de critères" (5). Below this is a "Description des critères" table with columns for "N°", "Code", "Libellé", "Poids", "Signe", "Unité", and "Description". The table contains five rows of criteria with pre-filled weights and signs. At the bottom of the interface are "Annuler" and "Suivant" buttons.

N°	Code	Libellé	Poids	Signe	Unité	Description
1	1		0.036	+		
2	2		0.192	+		
3	3		0.328	-		
4	4		0.328	+		
5	5		0.12	+		

Figure 13 : Création du problème de sélection du fournisseur logistique : Description

C'est un problème de type quantitatif et de catégorie problème de tri. Le problème dispose également de 5 critères dont les poids sont respectivement de c_1 à c_5 (0.036, 0.192, 0.326, 0.326, 0.12). Seul le critère c_3 est un critère à minimiser d'où le signe (-). Après cette étape vient l'étape Performances, il s'agit de l'énumération des différentes alternatives avec leur évaluation sur chaque critère formant ce que l'on appelle la matrice de performance. Ci-dessous la capture d'écran de cette étape Performance.

Problème de décision : Logistic provider selection

Description Performances Préférences Validation

1 2 3 4

2 Matrice de performances

Nombre d'alternatives Normalisées ?

Matrice de performances

Code ↕	Alternative ↕	1	2	3	4	5
A1	A1	60.0	0.4	2540.0	500.0	990.0
A2	A2	6.35	0.15	1016.0	3000.0	1041.0
A3	A3	6.8	0.1	1727.2	1500.0	1676.0
A4	A4	10.0	0.2	1000.0	2000.0	965.0
A5	A5	2.5	0.1	560.0	500.0	915.0
A6	A6	4.5	0.08	1016.0	350.0	508.0
A7	A7	3.0	0.1	1778.0	1000.0	920.0

Précédent Suivant

Figure 14 : sélection du fournisseur logistique : Performances

Nous disposons de 7 alternatives et chaque alternative est évaluée sur les 5 critères définis préalablement. À noter que les données ne sont pas normalisées, elles seront normalisées juste au début de la procédure d'agrégation. L'étape suivante concerne la prise en compte des préférences du décideur. Il s'agit d'un pré-ordre partiel entre des paires d'alternatives.

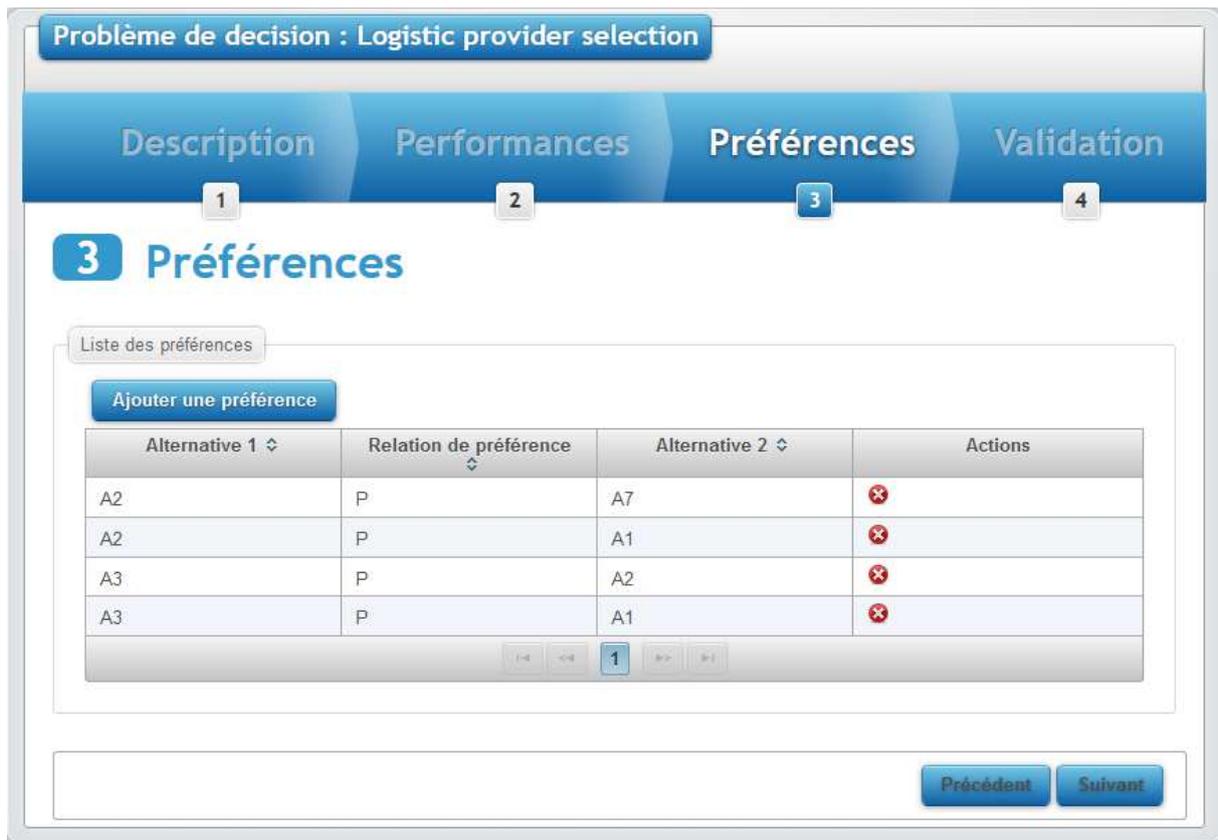


Figure 15 : Sélection du fournisseur logistique : Préférences

Des enquêtes antérieures ont montré que le décideur préfère A2 à A7, A2 à A1, A3 à A2 et A3 à A1. Ces préférences sont utilisées lors de la détermination de la mesure floue de l'intégrale de Choquet. Nous arrivons enfin à la phase de validation qui permet de stocker l'ensemble des données dans la base de données.

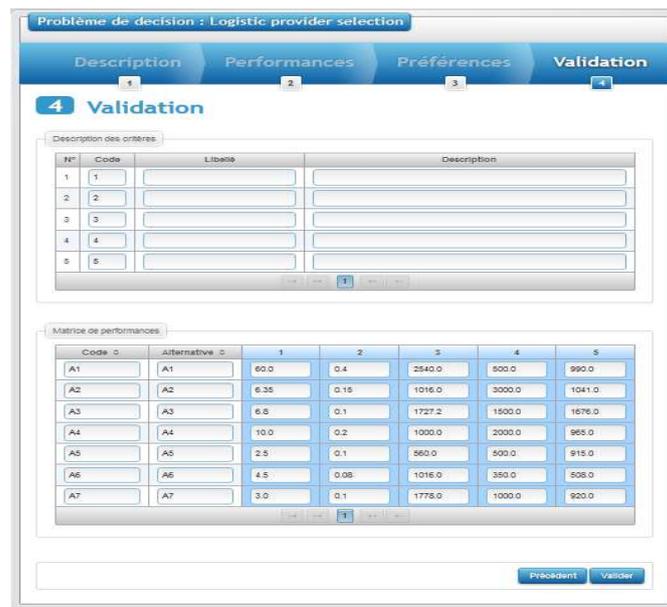


Figure 16 : sélection de fournisseur logistique : validation

Après cette étape de validation, l'ensemble des données sont enregistrées dans la base de données. Le problème de décision nouvellement créé, se trouve dans la liste des problèmes de décision. On peut à présent agréger les performances afin d'avoir un score global permettant de comparer les alternatives. Pour ce faire, nous avons deux possibilités : la première consiste à demander au système de nous proposer un opérateur d'agrégation en utilisant la méthode automatique. La deuxième consiste à sélectionner nous-même un opérateur d'agrégation que nous jugeons compatible avec notre problème de décision.

Nous allons dans un premier temps utilisé la méthode automatique sur ce problème de décision. Pour ce faire, il suffit de procéder comme suit :

Problèmes de décision									
					Operateur d'agrégation	Ajouter			
Code	Libellé	Type	Catégorie	Nombre de critères	Actions				
CC	Choosing a cook	Quantitative	PC	3	Editer	Supprimer	Critères	Alternatives	Agréger
CV	Choice of car	Quantitative	PC	5	Editer	Supprimer	Critères	Alternatives	Agréger
RB	Robot Selection	Quantitative	PT	5	Editer	Supprimer	Critères	Alternatives	Agréger
CV2	Choix du velo	Quantitative	PC	2	Editer	Supprimer	Critères	Alternatives	Agréger
CM	Choix moto	Quantitative	PC	3	Editer	Supprimer	Critères	Alternatives	Agréger
LPS	Logistic provider selection	Quantitative	PT	5	Editer	Supprimer	Critères	Alternatives	Agréger
CP	Choix d'un plombier	Quantitative	PC	4	Editer	Supprimer	Critères	Alternatives	Agréger
FM	Film	Quantitative	PT	5	Editer	Supprimer	Critères	Alternatives	Agréger
FLM	Film	Quantitative	PT	5	Editer	Supprimer	Critères	Alternatives	Agréger

Figure 17 : Liste des problèmes de décision, choix automatique

Pour lancer la procédure d'agrégation en utilisant le mode automatique, il suffit de se rendre dans la liste des problèmes de décision, ensuite sélectionner « Automatique » dans la liste déroulante des opérateurs d'agrégation et enfin cliquer sur le bouton agréger en face du problème de décision en question. En suivant ces étapes, voyons le résultat de la méthode automatique.



Figure 18 : sélection du fournisseur logistique, résultat de la méthode automatique(MOORA)

La méthode proposée de manière automatique pour la résolution de ce problème a été la méthode MOORA. Cette méthode a été proposée avec un degré de satisfaction « Bon ». Elle a été proposée car c'est une méthode quantitative et possède le meilleur degré de satisfaction sur les problèmes de tri. L'alternative qui arrive en tête est le A2 suivi de A4. L'alternative A6 arrive en dernière position juste après le A7.

Nous allons maintenant utiliser le choix manuel sur les autres opérateurs d'agrégation à savoir l'intégrale de Choquet, COPRAS, EDAS, WASPAS, TOPSIS et la somme pondérée.

Pour lancer une procédure d'agrégation en utilisant la méthode manuelle du choix de l'opérateur, il suffit de procéder comme suit :



Figure 19 : liste des problèmes de décision, choix manuel

Dans la liste déroulante des opérateurs d'agrégation, choisir l'opérateur d'agrégation souhaité pour cette procédure d'agrégation. En sélectionnant successivement l'intégrale de Choquet,

COPRAS, EDAS, WASPAS, TOPSIS et Somme pondérée nous obtenons les résultats suivants :

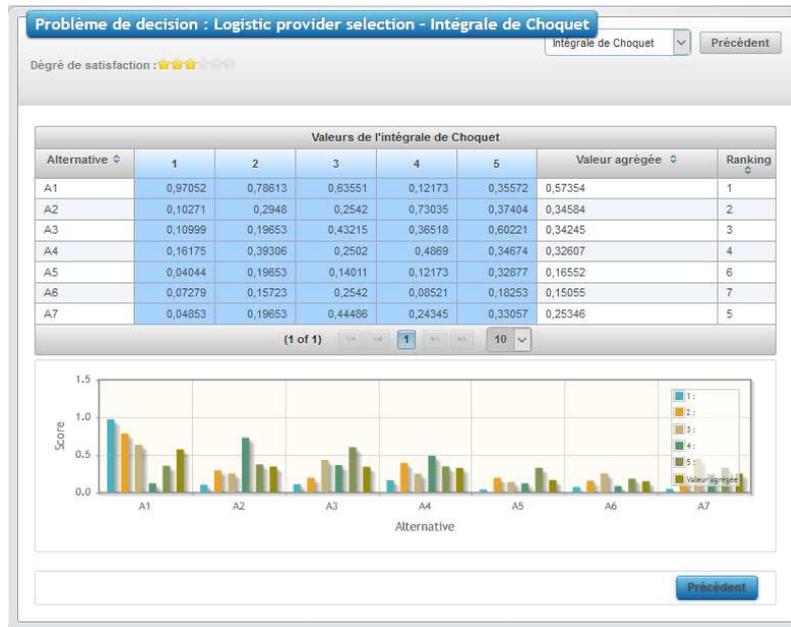


Figure 20 : sélection du fournisseur logistique : intégrale de Choquet

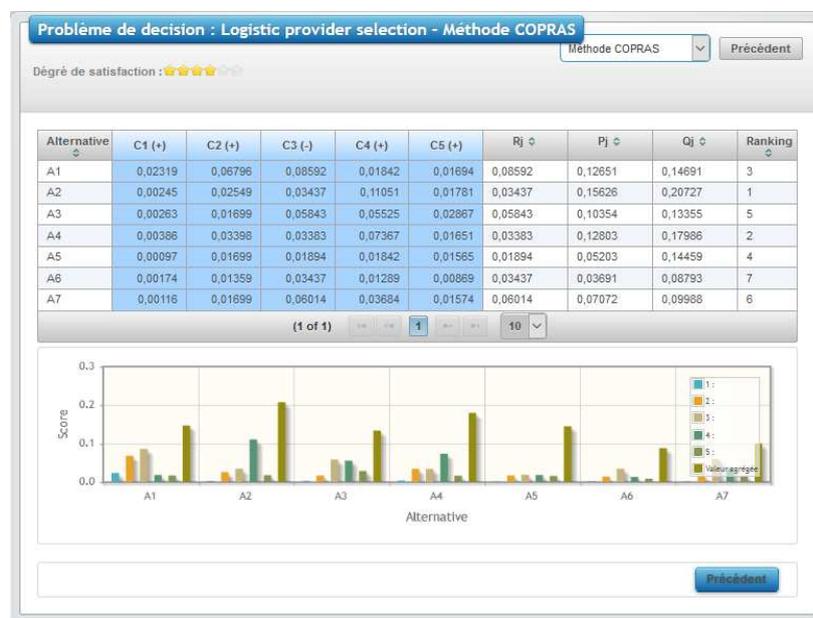


Figure 21 : sélection de fournisseur logistique : COPRAS

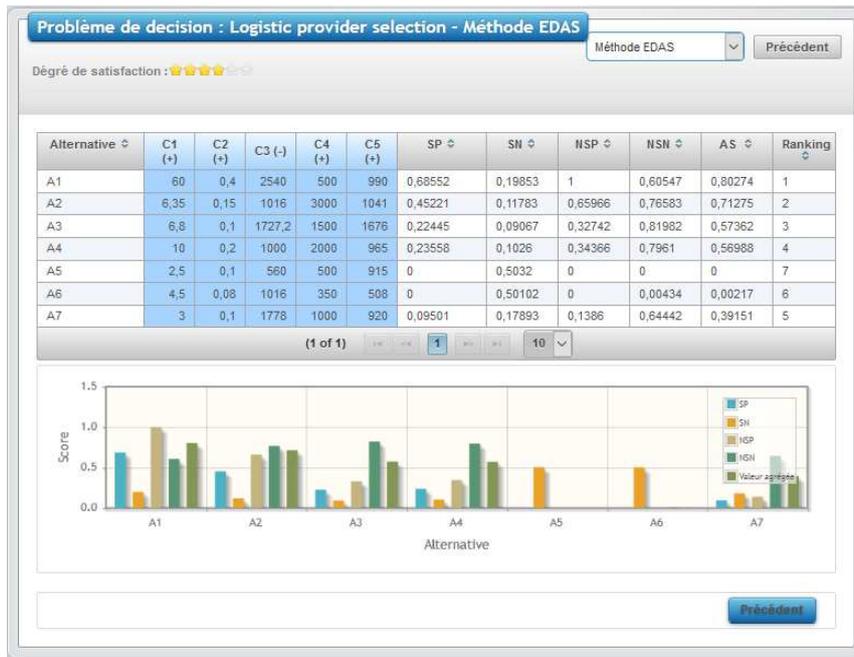


Figure 22 : sélection de fournisseur logistique : EDAS

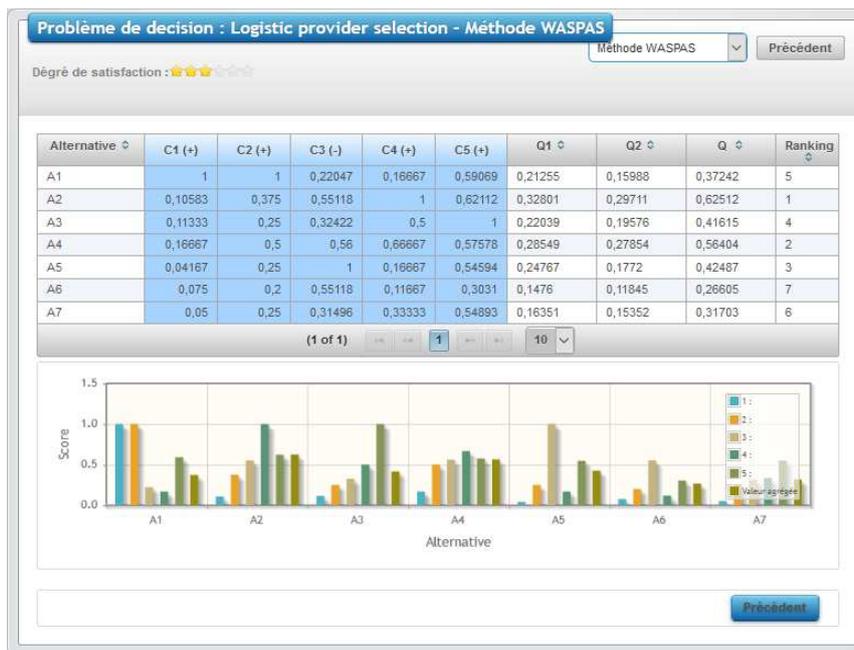


Figure 23 : sélection de fournisseur logistique : WASPAS

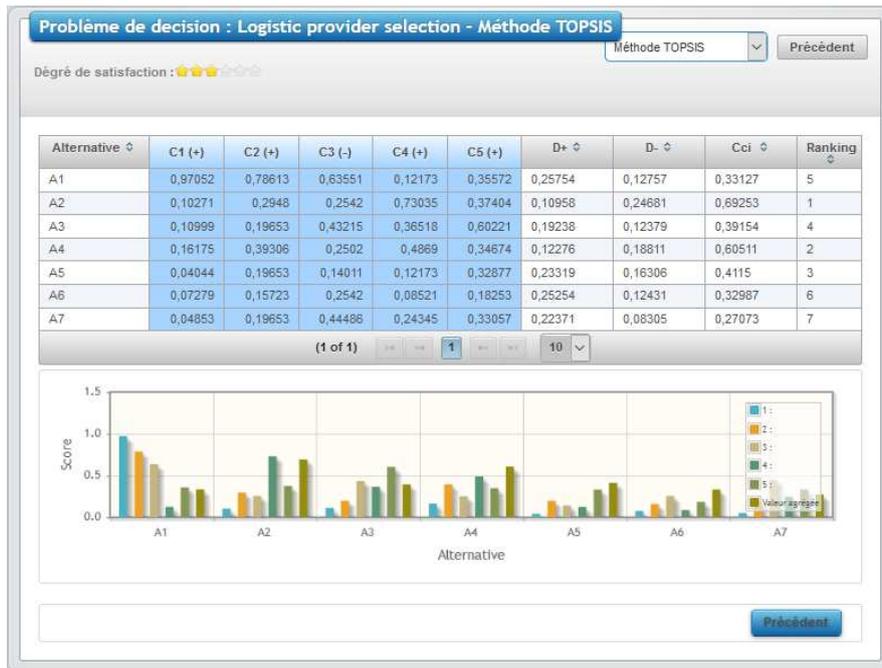


Figure 24 : sélection de fournisseur logistique : TOPSIS

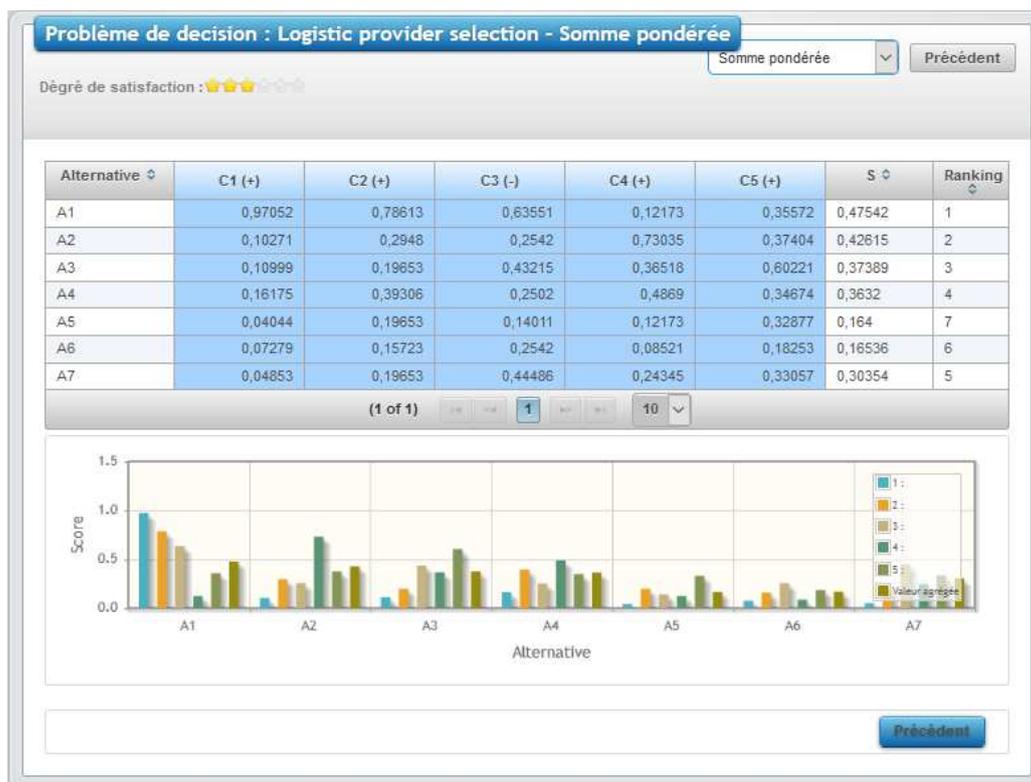


Figure 25 : sélection de fournisseur logistique : somme pondérée

Pour ces différentes méthodes sélectionnées manuellement tour à tour, seules deux méthodes (COPRAS et EDAS) ont un degré de satisfaction « Bon » égale à celui de la méthode automatique. Les autres ont des degrés de satisfaction inférieurs à celui de la méthode

automatique. Entre les méthodes MOORA (choisi automatiquement) et COPRAS, nous avons un coefficient de corrélation de Spearman Rho égal à 0.86. Ce qui représente un très bon niveau de similarité entre les résultats de ces deux méthodes. Par contre le coefficient de corrélation entre MOORA et EDAS est de 0.42 ce qui représente un mauvais score. Entre MOORA et intégrale de Choquet nous avons un coefficient de Spearman Rho égal à 0.53. Ce score peut être considéré comme moyen, mais il reste faible.

Table des illustrations

Figure 1 : Processus interactif de décision	28
Figure 2 : Mesure de similarité.....	64
Figure 3 : Système de recommandation d'Amazon	72
Figure 4 : PANDORA	73
Figure 5 : Last.fm	74
Figure 6 : Création d'un problème de décision	80
Figure 7 : Définition des critères d'un problème de décision	81
Figure 8 : Description des alternatives dans l'exemple choix du cuisinier	82
Figure 9 : Choix manuel d'un opérateur d'agrégation	83
Figure 10 : Choix automatique d'un opérateur d'agrégation	86
Figure 11 : Choix d'un cuisinier, méthode automatique	119
Figure 12 : Résultat de la sélection de robot	121
Figure 13 : Création du problème de sélection du fournisseur logistique : Description	134
Figure 14 : sélection du fournisseur logistique : Performances	135
Figure 15 : Sélection du fournisseur logistique : Préférences	136
Figure 16 : sélection de fournisseur logistique : validation	136
Figure 17 : Liste des problèmes de décision, choix automatique.....	137
Figure 18 : sélection du fournisseur logistique, résultat de la méthode automatique(MOORA)	138
Figure 19 : liste des problèmes de décision, choix manuel.....	138
Figure 20 : sélection du fournisseur logistique : intégrale de Choquet.....	139
Figure 21 : sélection de fournisseur logistique : COPRAS	139
Figure 22 : sélection de fournisseur logistique : EDAS	140
Figure 23 : sélection de fournisseur logistique : WASPAS	140
Figure 24 : sélection de fournisseur logistique : TOPSIS	141
Figure 25 : sélection de fournisseur logistique : somme pondérée	141

Table des tableaux

Tableau 1 : choix de la voiture du père de famille, évaluation des alternatives	38
Tableau 2 : Choix de la voiture du père de famille, matrice de performance	38
Tableau 3 : choix de la voiture du père de famille, poids des critères	39
Tableau 4 : choix de la voiture du père de famille, résultats.....	39
Tableau 5 : Choix de la voiture du père de famille, matrice de performance	40
Tableau 6 : Matrice de performance ordonnée et résultats	40
Tableau 7 : Classification des opérateurs évoqués.....	54
Tableau 8 : Propriétés des opérateurs évoqués	57
Tableau 9 : Liste des degrés de satisfaction	83
Tableau 10 : Classification des résultats possibles d'une recommandation d'un élément pour l'utilisateur	108
Tableau 11 : Exemple de valeur de coefficients de Kendall Tau-b	113
Tableau 12 : Résultats de l'utilisateur 1	117
Tableau 13 : Matrice d'utilité des 4 cuisiniers	118
Tableau 14 : choix du cuisinier, comparaison entre la méthode automatique et deux autres.	120
Tableau 15 : Exemple du problème de sélection de robot	120
Tableau 16 : comparaison des différentes méthodes sur la sélection de robot	122