



HAL
open science

Système réparti d'aide à la prise de décision distribuée et collective

Sana Nadouri

► **To cite this version:**

Sana Nadouri. Système réparti d'aide à la prise de décision distribuée et collective. Autre [cs.OH]. ISAE-ENSMA Ecole Nationale Supérieure de Mécanique et d'Aérotechnique - Poitiers; Université de Constantine, 2022. Français. NNT : 2022ESMA0019 . tel-03999063

HAL Id: tel-03999063

<https://theses.hal.science/tel-03999063>

Submitted on 21 Feb 2023

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

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



THÈSE

pour l'obtention du Grade de
**Docteur de l'École Nationale Supérieure de Mécanique et
d'Aérotechnique et l'Université Constantine 2**

Présentée par :

Sana NADOURI

**Systeme réparti d'aide à la prise de décision distribuée et
collective**

Directeurs de thèse : **Allel HADJALI et Zaidi SAHNOUN**

Co-encadrants de thèse : **encadrant**

Soutenue le 5 Décembre 2022
devant la Commission d'Examen

JURY

Président :

MAAMRI Ramdane

Prof. Université Constantine 2

Rapporteurs :

GHEDIRA-GUEGAN Chirine

Prof. Université Lyon 3

AKHROUF Samir

Prof. Université Mohamed Boudiaf

Examineurs :

MARSALA Christophe

Prof. Sorbonne Université

CHAOUI Allaoua

Prof. Université Abdelhamid Mehri



Remerciements

*Soyons reconnaissants aux personnes qui nous
donnent du bonheur ; elles sont les charmants
jardiniers par qui nos âmes sont fleuries.*

—Marcel Proust

Ce travail a été effectué entre le **L**aboratoire d'**I**nformatique **R**Epartie de l'université de Constantine 2 Abdelhamid Mehri, à la faculté des NTIC (nouvelles technologies de l'information et de la communication), au département de TLSI (technologies des logiciels et des systèmes d'information) et le **L**aboratoire d'**I**nformatique et d'**A**utomatique pour les **S**ystèmes à l'**U**niversité de **P**oitiers - ENSMA que je remercie.

Pour commencer, Un immense merci, tout à fait particulier, à mes parents et à mon frère pour leur soutien, leur patience et leur sacrifice. J'espère que je serais toujours une source de bonheur et de fierté pour eux.

Je tiens à témoigner mes remerciements et toute ma reconnaissance à mes directeurs de thèse : Prof. Allel HADJALI et Prof. Zaidi SAHNOUN pour m'avoir soutenu jusqu'à l'aboutissement de mes travaux. Je les remercie vivement pour leur aide efficace, leur expérience et leur entière disponibilité.

Je remercie très sincèrement les membres du jury d'avoir accepté d'évaluer ce travail. Je les remercie pour l'honneur qu'ils me font, pour leur participation scientifique ainsi que pour le temps qu'ils ont consacré à ma recherche.

Enfin, je tiens à remercier tous les intervenants qui ont contribué à la réalisation de cette thèse de près ou de loin.

Table des matières

Table des matières	vii
Liste des figures	ix
Liste des tableaux	xi
Introduction générale	1
I Préliminaires et états de l'art	5
1 Fondements des systèmes d'aide à la prise de décision (DSS)	7
1.1 Introduction	9
1.2 Présentation des Systèmes DSS	10
1.2.1 Définition	10
1.2.2 Architectures des SIAD et des DSS	10
1.2.3 DSS v.s. MIS	11
1.3 Systèmes DSS distribués (DDSS)	15
1.3.1 Définition	15
1.3.2 Systèmes DDSS et Multi-agents	17
1.3.3 Modélisation des DDSS avec l'approche SMA	22
1.3.4 Comparaison des modèles existants	24
1.4 Conclusion et Bilan	25
2 Paradigme Skyline	27
2.1 Introduction	29
2.2 Requêtes Skyline	30
2.2.1 Principe	30
2.2.2 Propriétés de base	31
2.3 Skyline de groupe	32
2.3.1 Définitions	32
2.3.2 Propriétés de base	34
2.4 Calcul du Skyline	35
2.4.1 Travaux sur le Skyline traditionnel	35
2.4.2 Travaux sur le Skyline de groupe	35
2.5 Conclusion et Bilan	39

II Contributions	41
3 Approche SMA pour la modélisation des DDSS	43
3.1 Introduction	45
3.2 Motivation	46
3.3 Aperçu du DDSS proposé	47
3.3.1 Structure générale de ADS^2	47
3.3.2 Structure interne de ADS^2	48
3.4 Modélisation du comportement du DDSS	49
3.5 Caractéristiques de l'approche ADS^2	51
3.6 Meta-model du framework ADS^2	51
3.7 Intégration du paradigme Skyline dans la prise de décision	52
3.8 Conclusion et Bilan	61
4 Approche distribuée et raffinée du modèle à base de SMA	63
4.1 Introduction	65
4.2 Définition de la problématique	66
4.3 Relaxation du Skyline de groupe	66
4.3.1 Dominance de groupe floue	66
4.3.2 Méthode RG-SKY	68
4.3.3 Propriétés de base	69
4.3.4 Calcul du RG-SKY	70
4.4 Étude expérimentale	74
4.4.1 Environnement expérimental	74
4.4.2 Tests expérimentaux	74
4.4.3 Résultats expérimentaux	75
4.5 Conclusion et Bilan	81
Conclusion générale	82
Bibliographie	85
.1 Annexe - Logique floue (ou Théorie des ensembles flous)	94
.1.1 Formes de la gradualité	95
.1.2 Sémantique du degré d'appartenance	95
.1.3 Concepts de base	95

Liste des figures

1	Graphe de relecture	3
1.1	Quelques générations des Systèmes d'information	9
1.2	Le processus de prise de décision dans les DSS	11
1.3	Architecture d'un Système d'Information Décisionnel [Kee80]	12
1.4	Topologies des DDSS selon [Ale02, NA14]	17
1.5	Architecture d'un agent réactif	18
1.6	Architecture d'un agent cognitif	18
1.7	Architecture d'un agent hybride	19
1.8	Architecture d'un agent BDI	20
1.9	Architecture SMA proposé par Ferber	21
2.1	Les points Skyline des hôtels de la conférence	30
2.2	Travaux connexes dans le contexte de calcul du Skyline de groupe	38
3.1	Architecture générale de ADS^2	47
3.2	Structure interne d'un DSS principal (IDS-MAS)	49
3.3	Comportement général du système proposé (ADS^2)	50
3.4	Méta modèle de ADS^2	51
3.5	le Framework de vérification de ADS^2	52
3.6	Processus général d'un DSS	54
3.7	Différentes étapes de l'approche proposée	57
3.8	Scénario d'étude de cas	59
3.9	Temps d'exécution de B2D	60
4.1	La fonction d'appartenance $MP_{G_i}^{(\gamma_{i1}, \gamma_{i2})}$	67
4.2	73
4.3	Résumé des différents tests expérimentaux	76
4.4	Temps d'exécution des données réelles et nombre de groupes Skyline relaxés renvoyés	77
4.5	Temps d'exécution des données synthétiques et nombre de groupes skyline détendus dans différents types de données	78
4.6	Synthetic correlated data with different dataset size	78
4.7	données synthétiques corrélées avec des valeurs dynamiques du groupe K (dimensions) et du groupe N (nombre de groupes)	79
4.8	données synthétiques corrélées avec des valeurs dynamiques du groupe K (dimensions) et du groupe N (nombre de groupes)	80
9	Représentation du prédicat "grand" dans le référentiel des tailles des personnes	94

Liste des tableaux

1.1	Comparaison entre les DSS et les MIS	14
1.2	Comparaison des travaux connexes	23
2.1	Symboles et leurs significations	32
2.2	Comparaison des algorithmes Skyline	33
2.3	Les données des joueurs	33
2.4	Les équipes Skyline (en utilisant la définition 1)	34
2.5	Comparaison des travaux liés au Skyline de groupe	37
3.1	Comparaison des travaux connexes avec notre travail	46
4.1	Exemple d'utilisation de Skyline pour trouver les meilleurs hôtels pour une conférence donnée	68
4.2	Spécifications des jeux de données réelles	74
4.3	Ensemble de paramètres (où N représente le nombre de groupes d'entrée)	75

Introduction générale

Contexte et problématique

Les systèmes d'aide à la décision (DSS pour Decision Support System) sont des systèmes composés de différents modules et utilisés par les cadres intermédiaires et supérieurs (d'une entreprise, par exemple) pour compiler des informations à partir d'un large éventail de sources afin de faciliter la résolution de problèmes et la prise de décisions. Afin de s'adapter au caractère distribué des organisations actuelles (aussi bien au niveau des données qu'au niveau des sites de décisions), les DSS ont évolué vers des systèmes plus généraux et efficaces, dits les systèmes distribués d'aide à la décision (DDSS pour Distributed Decision Support Systems). Les développements récents des systèmes DDSS ont conduit à de nouvelles situations de prise de décision. L'adaptabilité et le dynamisme, l'imprévisibilité, l'évolution et la distribution des environnements nécessitent d'améliorer le processus de décision. Les contextes militaires, gouvernementaux et médicaux sont des exemples de tels environnements. Les processus actuels des DDSS souffrent de plusieurs aspects tels que la mauvaise gestion de la masse d'information et la décentralisation des données, et la difficulté de générer une connaissance utile pour des prises de décisions complexes (destinées, par exemple, pour des groupes d'utilisateurs). Par ailleurs, les différentes entités d'un DDSS doivent être dotées d'un mécanisme de coopération afin d'améliorer les décisions fondées sur des données localisées sur des sites géographiquement distants.

Une des solutions pour gérer les aspects distribué et coopératif des systèmes DDSS est l'utilisation du paradigme Multi-Agent (SMA pour Systèmes Multi-Agents). L'étude des différentes conceptions des systèmes DDSS avec l'approche SMA, en s'appuyant sur un ensemble de critères de comparaison comme l'adaptabilité, l'apprentissage, l'objectif de décision, a permis d'identifier un ensemble de problèmes et limites ces systèmes. On peut citer le manque de tolérance aux fautes (présence d'un contrôle global), le manque d'adaptabilité, le mode d'apprentissage reste incomplet et il n'y a pas également de validation formelle de ces systèmes.

Principales contributions

Ce travail de thèse s'inscrit dans le cadre de l'amélioration des systèmes DDSS tant au niveau modélisation que exploitation des données pour la prise de décision. L'approche que nous avons proposée s'appuie sur la technologie agents et sur le paradigme Skyline. Les agents permettent de représenter la diversité des acteurs concernés par la décision ainsi que leurs interactions, d'une part, et de proposer des solutions robustes tout en s'adaptant à des environnements qui peuvent être imprévisibles et dynamiques, d'autre part. L'approche Skyline constitue un outil approprié pouvant aider l'utilisateur à prendre des décisions intelligentes en présence de données multidimensionnelles et lorsque différents, et souvent contradictoires, critères sont à prendre en compte.

Dans un premier temps, une revue critique est réalisée sur les différentes modélisations à base de MAS des DDSS, d'une part, et sur les travaux existants sur le paradigme Skyline dans les contextes de décision individuelle et de décision de groupe, d'autre part.

En termes de contributions, nous avons subdivisé notre solution en deux phases. La première contribution consiste à proposer une approche SMA pour la modélisation des DDSS. La spécificité de notre approche est sa généralité, son auto-adaptabilité, et sa rapidité de réponse grâce à l'intégration de l'opérateur Skyline. Nous avons supprimé le contrôle global et intégré les algorithmes de calcul de Skyline pour renforcer la fiabilité de la réponse. Il est à noter que l'approche intègre aussi une phase d'apprentissage pour l'identification des problèmes de décision similaires. Une validation formelle de l'approche a aussi été réalisée.

Dans un second temps, nous avons abordé le problème de Skyline de groupe à des fins de prise de décisions. Le modèle de Skyline proposé est centré utilisateur (tenant compte des préférences/profils des utilisateurs) et offrant des procédures de contrôle des résultats afin de garantir des décisions robustes et pertinentes. Le concept clé de ce modèle est une relation de préférence floue convenablement choisie. Cette contribution a porté sur la sémantique et aussi sur l'aspect algorithmique pour le calcul du Skyline de groupe relaxé. Enfin, une série d'expérimentations a été réalisée afin de valider nos contributions et montrer la fiabilité des approches proposées.

Structure de la thèse

Ce manuscrit est structuré en deux parties principales. La première partie dédiée à l'état de l'art, elle est constituée de deux chapitres. La seconde partie discute nos contributions, elle est composée aussi de deux chapitres. Une annexe est aussi intégrée au manuscrit.

Le chapitre 1 fournit une revue critique des différentes approches de modélisation des DDSS avec les systèmes multi-agents.

Le chapitre 2 présente le paradigme Skyline pour la prise de décision. L'accent est mis beaucoup plus sur le Skyline de groupe dédié aux décisions collectives. Il est à noter que dans ce chapitre les aspects liés à la sémantique et les aspects calculatoires sont explicitement abordés.

Le chapitre 3 est dédié à l'étude de l'approche SMA pour la modélisation des systèmes DDSS. Nous proposons le modèle $A^2D^2S^2$ ou tout court ADS^2 (Agent Approach for Distributed Decision Support Systems), une architecture multi-agent basée sur les modules DSS (Distributed Support Systems) standards, tels que : le décideur, l'interface utilisateur, le moteur de connaissances, la base de données et le module de traitement. L' ADS^2 est raffiné ensuite par un autre sous modèle basé sur le Skyline afin de renforcer la prise de décision interne du système globale.

Dans le chapitre 4, nous discutons notre modèle de Skyline de groupe étendu. L'idée est de développer une variante du Skyline de groupe relaxée. Cette relaxation permet d'augmenter le nombre de résultats de groupes Skyline et ainsi éviter le problème d'échec pour la prise de décision. Cette variante est dirigée par les préférences utilisateurs et offre un véritable contrôle des résultats retournés.

Nous clôturons ce manuscrit par une conclusion générale où nous résumons nos contributions et discutons les différents résultats obtenus. À la fin, nous dressons un ensemble de perspectives que nous jugeons intéressantes et prometteuses pour de futures travaux.

L'Annexe 1 introduit quelques notions de base de la logique floue utilisée comme cadre de modélisations des préférences utilisateurs.

Ci-dessous, nous donnons un graphe illustrant l'enchaînement de relecture des différents chapitres du manuscrit :

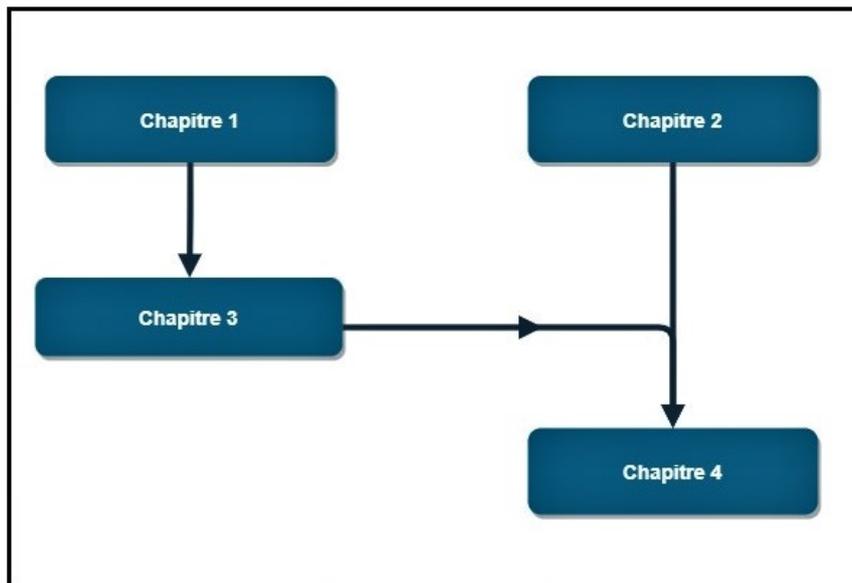


FIGURE 1 – Graphe de relecture

Liste des publications

Journal International

1. Sana Nadouri, Allel Hadjali, Zaidi Sahnoun (2022), **RG-SKY : A fuzzy group skyline relaxation for combinatorial decision making**, *Computer Science and Information Systems journal*, Vol. 19, No. 2, 887–912, 2022, <https://doi.org/10.2298/CSIS211020015N>.

Communications Internationales

1. Sana Nadouri, Allel Hadjali, Zaidi Sahnoun (2022), **Fuzzy Group Skyline Relaxation**, *Accepté dans la 31ème Conférences Francophones sur la Logique Floue et ses Applications, 20-21 Octobre, 2022, Toulouse, France.*
2. Sana Nadouri, Yassine Ouhammou, Zaidi Sahnoun, Allel Hadjali (2018), **Towards a multi-agent approach for distributed decision support systems**, *In Proc. of the 27th IEEE International Conference on Enabling Technologies : Infrastructure for Collaborative Enterprises (WETICE'2018), Paris, France, pp. 72-77.*
3. Sana Nadouri, Zaidi Sahnoun, Allel Hadjali (2018), **Using G-skyline to improve Decision-Making**, *In Proc. of the 3rd International Conference on Advanced Aspects of Software Engineering (ICAASE'2018), Constantine, Algérie, pp. 141-150.*
4. Sana Nadouri, Allel Hadjali, Zaidi Sahnoun (2019), **Vers une Décision Skyline Intelligente**, *Dans Actes de la 19ème Conférence Francophone sur l'Extraction et la Gestion des Connaissances (EGC'2019), Metz, France, pp. 371-372.*
5. Sana Nadouri, Allel Hadjali, Zaidi Sahnoun (2018), **Group Skyline Computation : An Overview**, *Dans Actes du 36ème Congrès INformatique des ORganisations et Systèmes d'Information et de Décision (INFORSID'2018), Nantes, France.*

Journées et séminaires

1. Sana NADOURI (2021), **Système d'aide à la prise de décision distribuée, approche Skyline et Multi-agents**, *LIB : Laboratoire d'Informatique de Bourgogne, Université de Dijon, France.*

Première partie

Préliminaires et états de l'art

Chapitre 1

Fondements des systèmes d'aide à la prise de décision (DSS)

Sommaire

1.1	Introduction	9
1.2	Présentation des Systèmes DSS	10
1.2.1	Définition	10
1.2.2	Architectures des SIAD et des DSS	10
1.2.3	DSS v.s. MIS	11
1.3	Systèmes DSS distribués (DDSS)	15
1.3.1	Définition	15
1.3.2	Systèmes DDSS et Multi-agents	17
1.3.3	Modélisation des DDSS avec l'approche SMA	22
1.3.4	Comparaison des modèles existants	24
1.4	Conclusion et Bilan	25

1.1 Introduction

Le domaine des systèmes d'information a connu une progression énorme. Chaque extension s'est appuyée sur ses prédécesseurs pour l'enrichir et l'améliorer. Son focus initial était le traitement des transactions (data processing systems), ce que nous appelons maintenant les Systèmes d'information de gestion (MIS pour Management information systems), dont l'objectif est la production de différents types de rapports pré-spécifiés contenant des informations jugées utiles aux gestionnaires [HJW94, Sim60].

Dans une expansion majeure au-delà des MIS, les systèmes d'information englobaient des systèmes conçus pour répondre aux Décideurs. Ce sont les systèmes d'aide à la décision (DSS pour Decision Support Systems). Un DSS peut être considéré comme une troisième génération des systèmes d'information, qui vise à essayer de découvrir ce qui se passerait si un certain nombre de décisions sont prises, aller au-delà des décisions ou de fournir automatiquement des suggestions d'administrateur. Ces systèmes ont commencé à émerger dans les années 70, ils se distinguent par leur capacités de satisfaire les besoins en terme de connaissances, effectuer la découverte de ces dernières et permettre une accessibilité directe et une personnalisation spécifique des fonctionnalités aux utilisateurs décisionnels, et/ou permettent l'apprentissage à partir d'expériences décisionnelles. Une autre expansion du monde SI est l'informatique organisationnelle (OC) qui désigne des Systèmes Informatiques qui facilitent des activités impliquant plusieurs participants. Ces organisations multipartites vont de groupes aux communautés à des entreprises complexes. Dans le cadre des systèmes OC, nous trouvons : le travail coopératif assisté par ordinateur, tout cela contribue et enrichit le contexte du DSS en lui donnant un concept de multi participant, la Figure 1.1 montre ces différents systèmes d'informations par couche complémentaire.

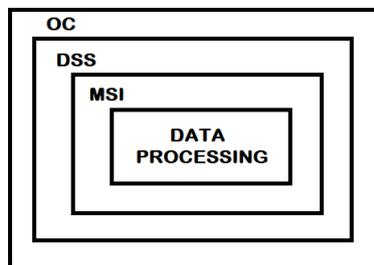


FIGURE 1.1 – Quelques générations des Systèmes d'information

Ce premier chapitre est organisé comme suit. Dans la Section 1 et 2 nous allons introduire les systèmes de prise de décision, leur processus interne, la structure générale d'un DSS et nous comparant le DSS avec les MIS. La Section 3 est consacrée aux Systèmes DSS distribués où nous allons voir le principe et où nous proposons leur définition. Nous allons ensuite comparer l'architecture DDSS avec les multi-agents et voir les avantages de ce paradigme SMA. La Section 4 conclut ce chapitre.

1.2 Présentation des Systèmes DSS

1.2.1 Définition

Les systèmes DSS sont des programmes informatiques utilisés par les cadres intermédiaires et supérieurs pour compiler des informations à partir d'un large éventail de sources afin de faciliter la résolution de problèmes et la prise de décisions.

Quant au Système Interactif d'Aide à la Décision (SIAD), il est défini de différente manière dans la littérature. Nous avons sélectionné la définition suivante "Un SIAD est un système ayant pour objectif d'aider les dirigeants d'entreprise à résoudre les différents problèmes liés à l'extraction de l'information et les aider dans la prise de décision". Le SIAD sert aussi d'outil d'analyse et de modélisation des données afin de créer des représentations à plusieurs dimensions de l'information transmise au sein de l'entreprise".

Les systèmes d'aide à la décision constituent un domaine essentiel des Systèmes d'Information (SI). Ces systèmes ont connu de multiples développements et continuent de croître sur différentes facettes.

1.2.2 Architectures des SIAD et des DSS

Le schéma de recherche des DSS (Figure 1.2), proposé depuis plus de 30 ans, a été doté d'une base solide pour la recherche et le développement continus des travaux de Keen et Scott Morton (1978) [Kee80], Bonczek, Holsapple et Whinston Et Carlson (1982) [HJW94], Bennett (1983) [EL90], et d'autres. Un système DSS soutient les décisions semi-structurées et non structurées et représente un champ multidisciplinaire composé de chercheurs issus des systèmes d'information de gestion (MIS), de la recherche opérationnelle, de l'intelligence artificielle (IA), des études organisationnelles et autres. Les progrès technologiques et les progrès de la recherche dans d'autres disciplines ont été rapidement adoptés au sein des composantes individuelles ou sous-systèmes des DSS.

De la même manière que les définitions varient en fonction des auteurs, il n'existe pas d'architecture standard pour un SIAD. En générale, il est constitué de plusieurs parties et sous-parties qui sont énumérées à travers l'architecture détaillée ci-dessous (Figure 1.3). Nous avons 5 composants de bases :

- un système de gestion des bases de données (SGBD)
- un système de gestion des modèles (MBMS)
- un moteur de connaissances (KE)
- une interface utilisateur
- l'utilisateur

Étape de prise de décision :

Cette étape est réalisée selon la situation considérée :

- Quand le problème est bien défini → la Décision est programmée
- Quand le problème est non structuré → la Décision sera basée sur l'information collectée

Comme nous allons le voir par la suite, les MIS en générale appartiennent à la première catégorie contrairement aux DSS qui nécessitent une adaptation de contexte basée sur l'information.

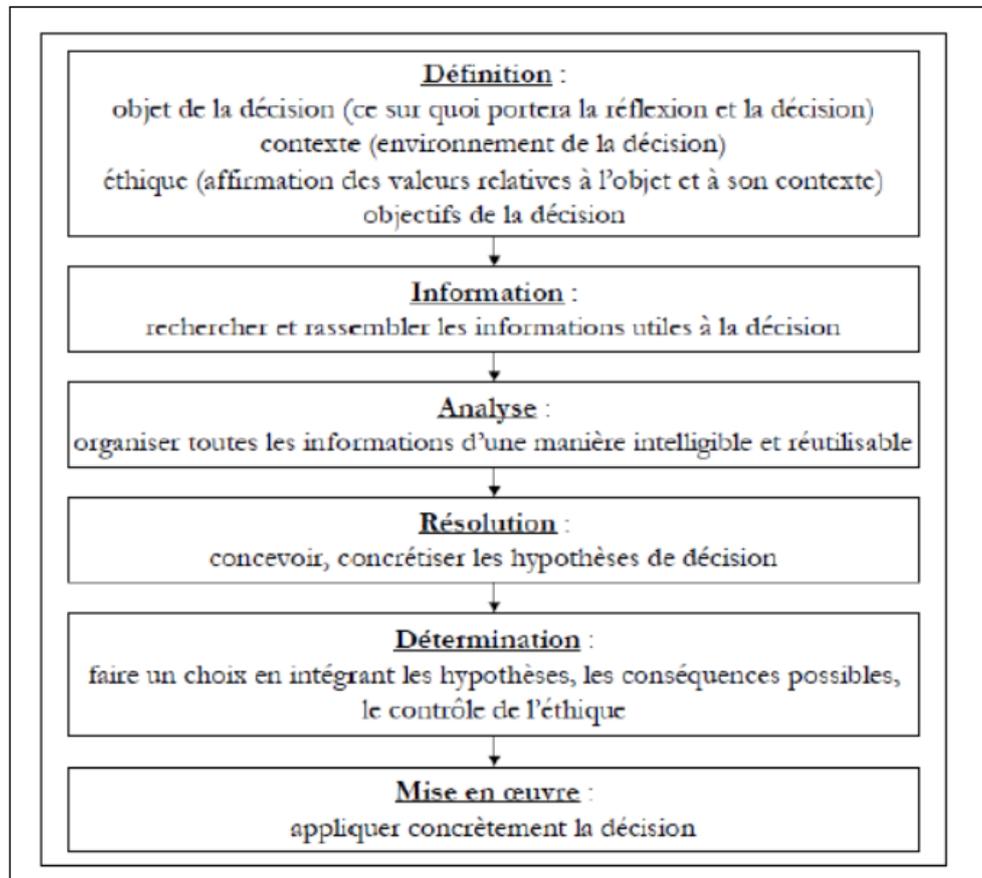


FIGURE 1.2 – Le processus de prise de décision dans les DSS

1.2.3 DSS v.s. MIS

Comparaison générale

La première utilisation du terme système d'aide à la décision était dans l'article de Goran et Scott-Morton (1971) [GSM71]. Ils ont soutenu que les systèmes d'information de gestion portaient principalement sur des décisions structurées et ont suggéré que les systèmes d'information de soutien pour les décisions semi-structurées et non structurées devraient être appelés "systèmes d'aide à la décision", MIS et DSS sont deux abréviations qui sont souvent entendues dans le domaine de la gestion des affaires. Ils diffèrent dans quelques aspects.

Il est intéressant de noter que le MIS est un type de lien qui aide à la communication entre gestionnaires de diverses disciplines dans une entreprise ou organisation. Dans l'ensemble, il joue un rôle très important dans la communication entre les gens d'entreprise. DSS de son côté est une amélioration de la notion MIS. Il est vrai que les deux diffèrent par leur orientation. DSS se concentre davantage sur le leadership. Il s'agit de la haute direction d'une entreprise offrant une vision innovatrice.

D'autre part, le MIS se concentre sur les informations recueillies. Par contre, Les experts en gestion indiquent que le DSS se concentre sur la prise de décision. MIS construit une planification de rapports de divers sujets liés à l'organisation qui aiderait les gestionnaires à prendre des décisions vitales relatives au fonctionnement de l'organisation.

Une des plus grandes différences entre MIS et DSS est que MIS se concentre sur l'efficacité opéra-

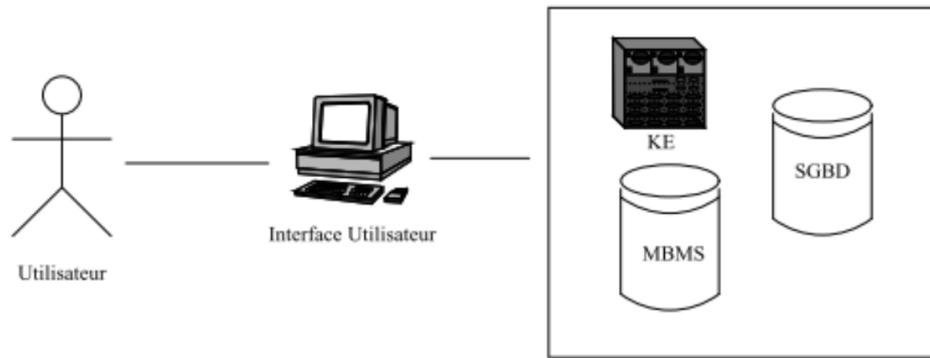


FIGURE 1.3 – Architecture d'un Système d'Information Décisionnel [Kee80]

tionnelle tandis que DSS se concentre plus sur la prise de décision efficace ou en d'autres termes, son but est d'aider la société à faire la bonne chose. Le flux d'information provient des deux côtés.

Dans le cas du DSS, le rapport peut être souple, alors que dans le cas de MIS, le rapport n'est généralement pas flexible. MIS est caractérisé par une entrée de grand volume de données, une sortie de rapports de synthèse et un processus caractérisé par un modèle simple. D'autre part DSS est présenté par une entrée de faible volume de données, une sortie d'analyse de décision et un processus caractérisé par un modèle interactif.

Les experts disent également que le MIS est un niveau primaire de prise de décision alors que le DSS est la partie principale de la décision. Une organisation devrait employer efficacement les deux systèmes.

Un DSS est conçu pour fournir une analyse efficace pertinente à des situations spécifiques pendant qu'un système d'information de gestion MIS- est conçu pour le traitement efficace de données ou d'information. Ils sont généralement utilisés par différentes personnes dans des situations différentes. Par exemple, les DSS sont utilisés par les gestionnaires et les autres décideurs dans les situations non structurées et semi-structurées et en grande partie dans des situations ad hoc, même si cela peut impliquer des décisions répétitives. Un DSS est utilisé dans les cas où un jugement de gestion individuel est nécessaire. Le MIS fournit des informations routinières typiquement sur une base continue dans un format standardisé. Les rapports peuvent être exécutés régulièrement ou être fournis sur demande. Les MIS sont souvent utilisés pour des fonctions courantes telles que le contrôle et la surveillance de la production, l'analyse financière et la recherche, en raison de la nature générale de l'information présentée.

À l'issue de cette introduction sur les deux systèmes, nous pouvons déduire un ensemble de caractéristiques propres à chacun d'eux.

1.2.3.1 MIS

- MIS intègre des flux d'information
- MIS répond aux besoins d'information de tous les niveaux de gestion
- Rapports axés sur la gestion
- MIS utilise une base de données intégrée

- Ne gère que les problèmes structurés
- Conçu pour supporter un ensemble d'applications
- Les MIS sont généralement des systèmes de reporting plus sophistiqués construits sur des systèmes de traite

Exemple : des décisions structurées soutenues par le MIS pourraient inclure la décision sur le niveau des stocks ou la tarification des produits.

1.2.3.2 DSS

- Interactif
- Ensemble intégré d'outils logiciels
- Produit de l'information pour appuyer le processus décisionnel
- Gère les problèmes semi-structurés et non structurés (décisions qui ne peuvent pas être décrites en détail)
- Peut être adapté à tout environnement décisionnel
- Aide le décideur
- Les problèmes DSS se caractérisent souvent par des connaissances incomplètes ou incertaines ou par l'utilisation de données qualitatives
- Interactif et convivial, peu d'aide informatique requise
- DSS inclut souvent des outils de modélisation, où différents scénarios alternatifs qui peuvent être modélisés et comparés

**Exemple : Les décisions d'investissement pourraient être soutenues par les DSS.
Comparaison détaillée**

Le tableau 2.2 montre les différences majeures entre ces deux types de systèmes.

Dimension	DSS	MIS
L'accent principal	L'important c'est la manipulation des données.	Le stockage des données est d'une grande importance
L'accès aux données	Les gestionnaires bénéficient d'un accès direct aux données.	Seul l'accès indirect aux données par les gestionnaires est fourni.
La nature de décision	Le gestionnaire possède la connaissance de la nature de la décision et de l'environnement décisionnel.	Gestionnaire MIS peut ne pas comprendre complètement la nature de la décision.
Concentration	Analyse et prise de décision	Traitement d'information
Utilisateurs typiques	Analystes, gestionnaires	Les cadres supérieurs
But	Efficacité	Opérationnel
Base de données	Spéciale	BD d'entreprise (limitée)
Prise en charge des décisions	Soutient la prise de décision semi-structurée ou non structurée	Soutien direct ou indirect, principalement des problèmes de routine structurés, en utilisant des opérations standard, la recherche et d'autres modèles
Type d'information	Informations pour soutenir des situations spécifiques	Rapports planifiés, Flux structuré, rapport d'exception
L'utilisation principale	Planification, organisation, et contrôle	Contrôle
Adaptabilité à un utilisateur individuel	Permet le jugement individuel, des capacités de type "what-if", utilise les somechoice comme style de dialogue	Habituellement aucun, normalisé
Graphiques	Partie intégrée dans de nombreux DSS	Souhaitable
Traitement de l'information	Les informations fournies par le MIS sont utilisées comme entrées au DSS	Des informations sont fournies à un groupe d'utilisateurs qui le manipulent
Informations détaillées complémentaires	Peut être programmé dans le DSS	Inflexibilité des rapports, ne peut pas obtenir les détails de soutien rapidement
Construction	Par les utilisateurs, seuls ou avec des spécialistes des départements SI	Par les vendeurs ou les spécialistes en SI
Nature des paquets informatiques	Grandes capacités de calcul	Application orientée, rapports de performance, Solides capacités de production de rapports, modèles statistiques, financiers, comptables et de gestion

TABEAU 1.1 – Comparaison entre les DSS et les MIS

1.3 Systèmes DSS distribués (DDSS)

La tendance actuelle à la globalisation a transformé la manière de prendre les décisions : les organisations sont maintenant bien souvent présentes sur plusieurs sites répartis à différents endroits de la planète. En conséquence, les données, les interfaces et les utilisateurs se situent bien souvent dans des lieux géographiquement distincts, ce qui rend l'environnement plus dynamique et peu prévisible. Les contextes militaires, gouvernementaux et médicaux sont des exemples de tel environnement. Les DSS distribués prennent en charge ces contextes et cette décentralisation de données.

L'architecture distribuée des DSS possède cinq caractéristiques principales [Ale02] :

1. Les ordinateurs sont dispersés dans une même organisation ou plusieurs organisations.
2. Les ordinateurs sont connectés par un système de communication de données.
3. Une base de données commune est partagée par tous, mais des bases de données supplémentaires peuvent exister.
4. Tous les ordinateurs sont centralement coordonnés avec une gestion de ressource d'information.
5. Les opérations d'entrée et de sortie sont effectuées dans les services utilisateurs.

Le principal avantage des systèmes centralisés est leur simplicité (temps de développement court). Cependant, le principal inconvénient est que tout est dans un seul endroit. Donc il n'y a pas de tolérance aux pannes et il souffre aussi d'une extensibilité médiocre. Par contre, les systèmes distribués sont en mesure d'améliorer cette situation.

1.3.1 Définition

Actuellement la prise de décision a beaucoup évolué. Ce qui engendre de nouvelles exigences à considérer. Cette nouvelle situation se caractérise par l'accroissement de la quantité d'informations qui circulent, ainsi que sa diversité. Aussi, la concurrence entre les organisations exige des prises de décisions efficaces et rapides. Les organisations se retrouvent éparpillées sur plusieurs sites (les données, les interfaces et les utilisateurs) géographiquement distincts. En conséquence, de nouvelles situations de prise de décision auxquelles sont confrontés les décideurs (managers) se déroulent dans des environnements dynamiques, changeant et distribués. Pour faire face à ces changements, les systèmes d'aide à la décision distribués sont apparus. Le DDSS est un DSS traditionnel mais avec un nouveau concept de topologie utilisée pour connecter les différents modules. En gardant certaines caractéristiques de bases telles que :

1. le DSS devrait orienter le comportement décisionnel des décideurs
2. le DSS doit toujours être aussi proche que possible des décideurs, prêt à être utilisé n'importe où, à tout moment.

Autres Caractéristiques d'un DSS

1. Un DSS distribué n'est pas nécessairement un système à données intensives
2. Dans un DSS réparti, deux unités de données qui ne sont pas sémantiquement liées peuvent toujours être physiquement stockés dans différents périphériques de stockage
3. Un DSS distribué tire partie de l'architecture globale distribuée d'Internet
4. Un DSS distribué peut survivre dans un réseau non fiable
5. Un DSS réparti améliore la mobilité
6. Un DSS distribué ne remplace pas les réunions face-à-face, il les améliore

Dans ce travail, nous considérons un DDSS comme :

Un ensemble de DSS qui communiquent dans un réseau dynamique afin de proposer un choix de prise de décision à des utilisateurs

1.3.1.1 Architecture des DSS distribués

Les DSS distribués sont définis selon 3 topologies différentes :

Architecture centralisée :

Dans une architecture centralisée, les différentes entités sont coordonnées par une entité hiérarchiquement supérieure (coordonnateur). La prise de décision pour un niveau est centralisée dans le centre coordinateur qui possède la totalité des données nécessaires transmises par les entités d'hierarchie qu'il supervise. Les systèmes centralisés sont caractérisés par :

- Simples à développer qui est un facteur important pour le développement de DSS (temps de développement court)
- La topologie centralisée permet un contrôle aisé sur les données et les utilisateurs
- Assure une cohérence des choix et n'illustre pas le caractère distribué des décisions
- Noyau centrale fragile et aucune tolérance aux fautes
- Dépendance du lieu et faible extensibilité (évolution limitée)
- Cette architecture repose sur une source centrale de synchronisation

Architecture décentralisée :

Contrairement à l'architecture centralisée, cette architecture permet d'illustrer la distribution des décisions. Les systèmes décentralisés ont des caractéristiques opposées à celles des systèmes centralisés, ils sont :

- Tolérants aux fautes (dans une certaine mesure), la réplication des composants importants entre les différents nœuds permet de l'assurer indépendamment du lieu et cette architecture est fortement extensible c-à-d l'ajout de nouveaux composants sans devoir suspendre le fonctionnement de l'ensemble, elle demande la mise au point de méthodes spécifiques de gestion pour assurer une cohérence globale des différents choix
- la difficulté à connaître et à maîtriser l'état du système

Architecture hybride :

Les topologies hybrides restent fortement basées sur des architectures centralisées traditionnelles. Afin de résoudre les problèmes de ces architectures, la topologie hybride isole quelques composants jugés (centraux) à l'architecture dans un anneau. De cette manière, ces composants sont clairement identifiés et donc plus faciles à administrer. Afin de ne pas mettre en danger l'ensemble du système en cas de panne de l'un de ces composants centraux, une certaine redondance des fonctionnalités et des connections est prévue :

- Elle traite les cas relatifs à la synchronisation de données et la réplication de données ou équilibrage de la charge
- Elle est peu utile dans des situations extrêmes
- Il est difficile d'identifier dans ce type d'architecture les composants jugés centraux de distribués lors de la conception

La figure 1.4 illustre ces différentes architectures :

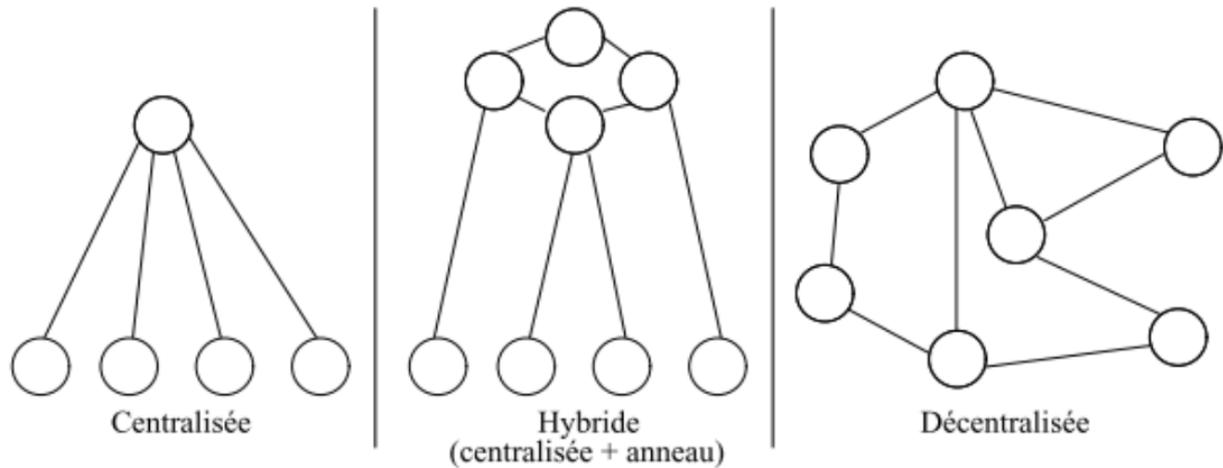


FIGURE 1.4 – Topologies des DDSS selon [Ale02, NA14]

1.3.2 Systèmes DDSS et Multi-agents

Un des développements récents des DDSS est l'utilisation des systèmes multi-agents pour gérer leurs aspects distribué et coopératif. Avant d'expliquer le bénéfice des multi-agents, nous commençons tout d'abord par rappeler le principe de cette technologie.

1.3.2.1 Système Multi-Agents (SMA)

Selon Ferber et d'autres auteurs, un agent est une entité physique ou virtuelle

- (a) Qui est capable d'agir dans un environnement,
- (b) Qui peut communiquer directement avec d'autres agents,
- (c) Qui est lié par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voir de survie, qu'elle cherche à optimiser),
- (d) Qui possède des ressources propres,
- (e) Qui est capable de percevoir (mais de manière limitée) son environnement,
- (f) Qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),
- (g) Qui possède des compétences et offre des services,
- (h) Qui peut éventuellement se reproduire,
- (i) Dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit [HAF09, FW99, LD12].

Presque toutes les définitions en l'intelligence artificielle (IA) de la notion d'agent abordent une notion essentielle : l'autonomie (c'est à dire, l'agent doit pouvoir prendre des initiatives et agir sans l'intervention d'un tiers (humain ou agent) et contrôler ses propres actions ainsi que son état interne.)

Types d'agents

Les agents peuvent être de deux types [FW99, OHB08] : des agents cognitifs dits "intelligents" et des agents réactifs dits "moins intelligents". A ces deux types d'agents, s'ajoute l'architecture hybride et l'architecture BDI.

— **Agents réactifs :**

L'approche réactive se base sur le concept d'émergence. Elle avance par ailleurs, l'idée qu'il n'est pas nécessaire que chaque agent soit personnellement "intelligent" pour parvenir à un comportement intelligent de l'ensemble. Les agents réactifs sont les plus faciles à concevoir. Ce type d'agent n'a pas de représentation de son environnement ou des autres agents, son comportement est décrit par des règles simples du type stimulus/réponse. Il ne tient pas compte du passé et ne planifie pas le futur. Il est difficile de voir comment un agent purement réactif peut apprendre de son expérience et améliorer ainsi ses performances.

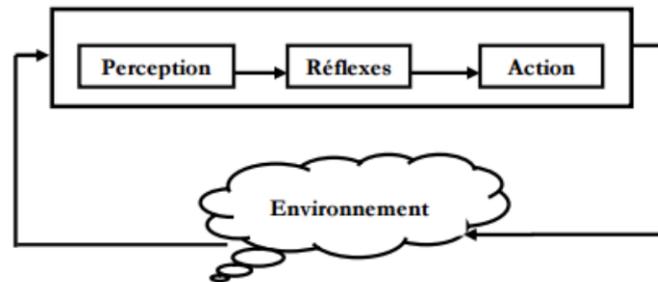


FIGURE 1.5 – Architecture d'un agent réactif

— **Agents cognitifs :**

C'est un type d'agents plus complexe doté de capacités de raisonnement importantes et disposant d'une représentation de son environnement. Chaque agent est muni d'une base de connaissances nécessaire à la réalisation de sa tâche ainsi qu'à la gestion des interactions avec son environnement. Pour atteindre leurs buts, ces agents utilisent l'expérience qu'ils ont acquise pour prendre des décisions, ils suivent le plan qu'ils ont établi jusqu'à son terme avant d'entreprendre une nouvelle tâche. L'agent cognitif est intéressant individuellement et collectivement et reprend les différentes méthodes de l'intelligence artificielle auxquelles vient s'ajouter un comportement social.

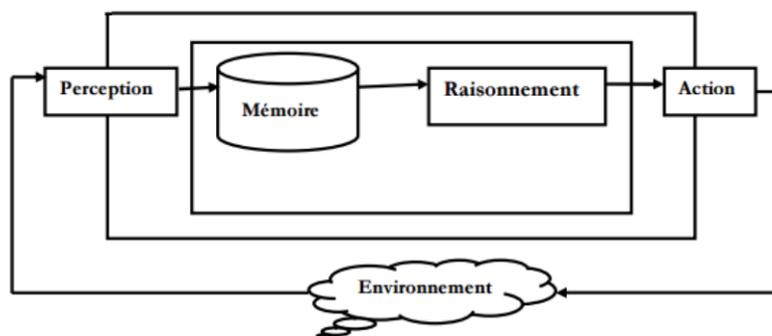


FIGURE 1.6 – Architecture d'un agent cognitif

— **Architecture hybride :**

Nous pouvons constater que les agents réactifs peuvent bien convenir pour certains types de problèmes et moins bien pour d'autres. De même, pour les agents cognitifs. Ce qui à

donner naissance à une autre approche qui combine les deux afin d'obtenir une architecture hybride. L'architecture hybride est en quelque sorte un compromis entre un agent purement réactif et un agent purement cognitif. Cette vision permet de concilier les capacités intéressantes des deux types d'agents précédents. La structure d'un agent hybride peut être divisée en trois couches. Une couche réactive qui va s'intéresser au traitement des données provenant de capteurs, une couche intermédiaire raisonnant sur les connaissances de l'agent à propos de son environnement et enfin, une couche supérieure prenant en considération les autres agents du système dans la phase de raisonnement.

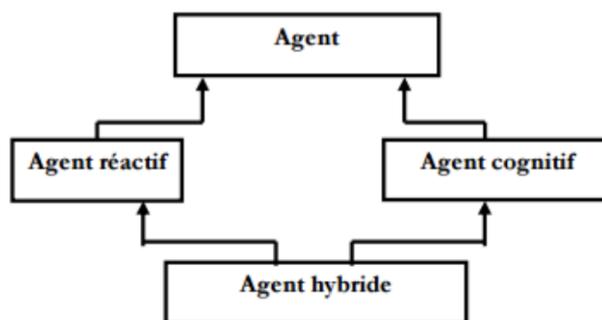


FIGURE 1.7 – Architecture d'un agent hybride

— Architecture BDI (Beliefs Desires and Intentions) :

Les agents BDI sont inspirés des travaux relatifs aux raisonnements pragmatiques, c'est-à-dire le processus de décision permettant de sélectionner les actions à effectuer pour atteindre ses objectifs. C'est une architecture très répandue, elle est basée sur un raisonnement prenant en compte les états mentaux, pour les agents ayant un état mental reposant sur :

- Les croyances : ce que l'agent connaît de son environnement.
- Les désirs : les états possibles vers lesquels l'agent peut vouloir s'engager.
- Les intentions : les états vers lesquels l'agent s'est engagé ou a engagé ses ressources.

Un agent BDI doit mettre à jour ses croyances avec les informations qui lui parviennent de son environnement, décider quelles options lui sont offertes, filtrer ses options afin de déterminer de nouvelles intentions et poser ses actions au vu de ses intentions.

Systèmes multi-agents

Pourquoi cherche-t-on à créer des intelligences collectives? Pourquoi tout simplement vouloir distribuer l'intelligence?

La complexité des systèmes serait la première raison, ces systèmes font intervenir un grand nombre de sous-systèmes, comprenant de nombreuses fonctionnalités, interagissant avec plusieurs spécialistes humains, qui se trouvent souvent repartis naturellement dans l'espace physique [FW99].

Un système SMA est composé des éléments suivants :

1. Un environnement E , c'est-à-dire un espace.
2. Un ensemble d'objets O . Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents.

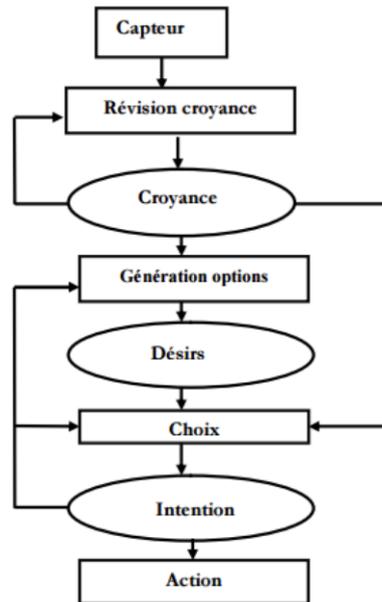


FIGURE 1.8 – Architecture d'un agent BDI

3. Un ensemble A d'agents, qui sont des objets particuliers, ils représentent les entités actives du système.
4. Un ensemble de relations R qui unissent des objets (et donc des agents) entre eux.
5. Un ensemble d'opérations Op permettant aux agents de A de percevoir, produire, consommer, transformer et manipuler des objets de O .
6. Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers [FW99]

Les caractéristiques principales d'un SMA sont [Pot10, Bri09] :

1. chaque agent possède des informations ou des capacités incomplètes pour résoudre le problème et a donc un point de vue limité.
2. il n'existe pas de contrôle global du système.
3. les données sont décentralisées.
4. le calcul est asynchrone.

L'architecture SMA (voir la figure 1.9) est définie principalement par un ensemble d'agents dans un environnement qui communique de différentes manières selon le problème traité [FW99, Gau07].

La force du paradigme agent provient de la flexibilité et de la variété des interactions entre les agents. Ces derniers coopèrent, négocient, se coordonnent les uns avec les autres, tant pour poursuivre leurs propres buts que pour atteindre un objectif commun. Nous citons trois modes d'interactions :

Coopération

La coopération est nécessaire car un agent ne peut atteindre ses buts sans être aidé, les agents ont une connaissance partielle des autres agents du système, mais ils doivent être informés sur leurs compétences, et les tâches que ces derniers sont en train de réaliser.

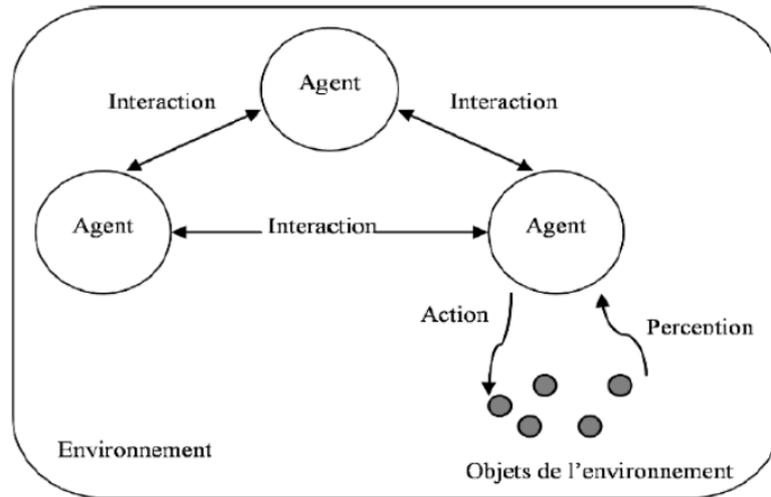


FIGURE 1.9 – Architecture SMA proposé par Ferber

Coordination

Pour garantir la cohérence globale d'un SMA, il est nécessaire que les agents coordonnent leur différentes actions. La coordination d'action est l'ensemble des actions supplémentaire qu'il est nécessaire d'accomplir dans un environnement multi agent et qu'un seul agent poursuivant les mêmes buts n'accomplirait pas. Le but de la coordination est de trouver, parmi un ensemble de comportements d'agents qui interagissent, une collection de comportements réalisant d'une façon satisfaisante les objectifs les plus importants des agents.

Négociation

Permet de résoudre des conflits qui pourraient mettre en péril des comportements coopératifs. La négociation est le processus par lequel plusieurs individus prennent une décision commune. Les participants expriment d'abord des demandes contradictoires, puis ils essaient de trouver un accord par concession ou par la recherche de nouvelles alternatives.

1.3.2.2 Avantages des SMA

Les systèmes multi-agents sont particulièrement appropriés pour les DDSS. En effet, la technologie agents a remplacé les modèles traditionnels pour la conception d'applications réparties et coopératives et/ou collaboratives car selon [NA14, BdNdL16] les multi-agents permettent de représenter la diversité des acteurs concernés par la décision, afin de parvenir à un accord qui satisfait ces différents acteurs (par les mécanismes de négociation propres aux SMA). Ils sont connus pour modéliser des systèmes complexes à partir des entités qui les composent et de leurs interactions et donnent un niveau d'abstraction très élevé qui permet d'exprimer l'ensemble des DSS en termes d'agents autonomes et souvent hétérogènes qui jouent des rôles (aide à la prise de décision) dans un environnement commun et rendent des services ou réalisent une fonction cohérente dans un cadre collectif. Un SMA gèrent l'augmentation de la masse d'informations à traiter et à stocker qui sont dus à cette distribution et répondent aux applications qui sont physiquement et fonctionnellement distribuées comme le cas des DDSS, par une conception modulaire de ces dernières. L'architecture multi-agents permet une maintenance plus facile du système et l'amélioration du traitement est localisée en général au niveau d'un agent(s). Nous pouvons citer aussi d'autres avantages de ce type de système :

- La décomposition et la répartition des connaissances et des mécanismes de traitement dont l'unité de base est l'agent.
- L'approche par les systèmes multi-agents propose des solutions robustes et capables de s'adapter dans des environnements qui peuvent être imprévisibles et dynamique comme le contexte DDSS.
- Limitations des approches informatiques classiques : statiques, homogènes, interfaces rigides, objets/composants sans initiative propre. f
- L'architecture SMA est plus efficace lorsqu'une architecture centralisée ne peut être mise en place du fait de l'indépendance des acteurs qui la compose.
- Le dynamisme du contrôle de résolution d'un problème distribué caractérisée par l'organisation dynamique et l'affectation des tâches modifiables en cours de résolution.
- L'aptitude à traiter des problèmes simultanés et potentiellement corrélés avec des optimisations éventuelles.
- L'adaptabilité et la possibilité d'apprentissage des agents qui leur confèrent la capacité de résister à des environnements évolutifs et/ou instables.

Domaines d'applications des systèmes multi-agents : Les systèmes multi-agents étant issus du domaine de l'intelligence artificielle distribuée, ils permettent de modéliser des applications où une modélisation classique est inadéquate, et où le système est souvent naturellement distribué. Si un problème n'est pas naturellement distribué, on peut choisir de le distribuer pour des raisons de simplification. Un aspect intéressant pour l'application de la notion d'agents est précisément que ce concept permet d'une façon naturelle de modéliser une gamme de systèmes complexes avec des besoins divers. Les systèmes multi-agents ont été appliqués à plusieurs domaines, tel que :

Les systèmes d'aide à la décision : sont utilisés dans de nombreux domaines. Ils ont pour objectif d'aider le décideur dans sa tâche en lui fournissant tous les éléments pertinents pour une prise de décision. Cela consiste très souvent à extraire de l'information depuis de multiples sources et à la traiter. Les systèmes multi-agents sont bien adaptés pour ce genre de traitement de l'information qui peut revêtir diverses formes et provenir de diverses sources. En effet, il faut pouvoir traiter efficacement l'ensemble des éléments obtenus pour les présenter à l'utilisateur. Aussi, l'approche à base d'agents permet la mise en œuvre des principes de négociation et de coopération qui sont très adaptés à ce domaine. Dans le cas des systèmes d'aide à la décision distribués. Néanmoins, la problématique multi-agents est assez complexe dans ce contexte d'agent. En effet, les systèmes multi-agents classiques, la composante sociale et les modèles afférents portent sur les agents eux-mêmes, dans ce cas, du fait de la finalité d'assistance à un processus social humain, les agents vont manipuler deux types de représentations sociales distinctes mais interdépendantes : un modèle de la société d'agents et un modèle de la société d'acteurs.

1.3.3 Modélisation des DDSS avec l'approche SMA

Plusieurs recherche et travaux se sont penchés sur la modélisation des systèmes DDSS par des SMA. Dans ce qui suit, nous avons sélectionné un ensemble de ces travaux que nous allons étudiés en détails

Une première classification sera par domaine et spécificité de l'approche proposée :

1.3.3.1 Approches DDSS générales :

Les approches DDSS générale qui ne dépend d'aucun domaine sont les suivants :

- A multi-agent framework for group decision support system : application to a boiler combustion management system (glz) Rédigé par : noria taghezout, abdelkader adla, pascale

		Distributed Decision Support Systems Criteria											
Approaches	1	2	3	4	5	6	7	8	9	10	11	12	
Part I	a	No	Quality	Is defined	No	Yes	No	Yes	Is not defined	No (referential)	General	Yes	Yes
	b	No	Quality	Is not defined	No	Yes	No	Yes	Is not defined	No (referential)	General	No	Yes
	c	No	Quality (with a limited time)	Is not defined	No	Yes	No	Yes	Is not defined	No (referential)	General	No	Yes
	d	Yes	Time	Is not defined	Yes	Yes	No	Yes	Is not defined	No	General	Yes	Yes
	e	Yes	Quality	Is defined	No	Yes	No	Yes	Is not defined	Yes	Specific (marketing)	No	No
	f	Yes	Quality	Is defined	Yes	Yes	Yes	Yes	Is not defined	No	Specific (brain tumour)	Yes	Yes
	g	Yes	Quality	Is not defined	Yes	Yes	No	Yes	Is not defined	No	Specific (sport event)	No	No

1. Adaptability, 2. Decision objective (quality/time), 3. Internal DSS model, 4. predictability, 5. Learning, 6. Reliability, 7. Interoperability, 8. Formal architecture, 9. Distributed data, 10. Domain (general/specific), 11. Fault tolerance, 12. Implementation

TABLEAU 1.2 – Comparaison des travaux connexes

zaraté (2009) [TAZ09a]

- Facilitating virtual group decision making (MODELE MULTI-AGENTS POUR L'AIDE A LA DECISION DE GROUPE) Rédigé par : Abdelkader Ould Mahraza, Djamilia Bouhalouan, Abdelkader Adla (2016) [MBA16a]
- Multi-Agents Model for Web-based Collaborative Decision Support Systems Rédigé par : abdelkader adla, bakhta nchet, Abdelkader Ould Mahraza (2012) [ANOM12a]
- Concevoir un système d'aide à la décision basé sur un système multi-agent anytime (Des systèmes multi-agents Anytime pour la conception de systèmes d'aide à la décision) Rédigé par : Claude Duvallet , Bruno Sadeg (2004) [DS04b]
- Architecture and design of distributed decision support system based on agent grid Rédigé par : jia-yu chi, ling sun, xue-guang chen, jin-long zhang [CSCZ05]
- Knowledge-intensive collaborative decision support for design process, part i : a hybrid decision model and multi-agent framework Rédigé par : xuan f zha, ram d sriram, marco gero fernández and farrokh mistree [ZSF⁺05]

1.3.3.2 Approches DDSS spécifiques :

D'autres approche spécifique ont été étudié afin de trouver les différents modèle SMA proposé tout type de DDSS, voici quelques travaux.

- Aménagement du territoire et SIG :

- Conception d'un système interactif d'aide à la décision collective en localisation spatiale (un SIAD de groupe en aménagement du territoire) Rédigé par : oufella Sarah (2009) [OHB09]
- Modèles Multi-Agents pour l'Aide à la Décision et la Négociation en Aménagement du Territoire (formelle) [Fer97b]

- Chaines logistique :

- Outils d'aide à la décision pour la prise de commandes imprévues Rédigé par : Smail Khouider [Kho08]
- Architecture d'aide a la décision distribuée et de simulation proactive dans les chaînes logistiques : une approche multi-agent [Nfa08]
- Conception et développement d'un système multi-agent d'aide à la décision pour la gestion de production dynamique Rédigé par : Noria TAGHEZOUT (2011) [Tag11]

1.3.3.3 Autres travaux :

Le reste des travaux à été regroupés dans la liste suivante :

- A distributed marketing decision support system based on multi-intelligent-agent Rédigé par : wen-guo ai, jie sun, hui li [ASL04a]
- Agent-based distributed decision support system for brain tumour diagnosis and prognosis “healthagents” (+ implementation) Rédigé par : horacio gonzález-véleza , mariola mierb , carles arúsc , bernardo celdad , sabine van huffele , paul lewisf , andrew peetg , monsterrat roblesh [GVMA⁺06a]
- Design of distributed support decision-making system model based on multi-agent technique for sporting event Rédigé par : song yingdong [Yin14a]
- Multi-agent systems and distributed constraint satisfaction for decision support in marine ecosystem management Rédigé par : ngisiange , rimiru , okeyo , wambiji ,aura [ANOW16]
- Distributed decision support system for airport ground handling management using wsn and mas Rédigé par : pablo garcia ansola, javier de las morenas, andres garcia, javier otamendi [ADLMGO12]

1.3.4 Comparaison des modèles existants

Au cours de la dernière décennie, les systèmes distribués d'aide à la décision (DDSS) ont reçu de plus en plus d'attention. Un certain nombre d'études ont été proposées. Nous avons sélectionné quelques travaux les plus intéressants de la partie précédentes sur la modélisation du DDSS à l'aide de SMA. Nous les avons classées en deux catégories : génériques et spécifiques. Le premier ne dépend pas d'un domaine d'application comme [TAZ09b] (a), [MBA16b] (b), [ANOM12b] (c), [DS04a] (d). La deuxième catégorie d'approches est appliquée dans un domaine d'étude spécifique, par exemple le marketing [ASL04b] (e), le diagnostic et le pronostic des tumeurs cérébrales [GVMA⁺06b, LAEM⁺07] (f), l'événement sportif [Yin14b] (g).

La synthèse de notre comparaison des travaux connexes s'appuie sur des propriétés et des paramètres de base des systèmes DDSS.

Nous avons étudié les différentes conceptions de DDSS avec l'approche Multi-agents en proposant un ensemble de critères de comparaison comme l'adaptabilité, l'apprentissage, l'objectif de décision, etc. qui nous ont permis d'identifier un ensemble de problèmes et limites qui seront utilisés dans le chapitre 3 pour la proposition d'un modèle plus raffiné et fiable. Le tableau 1.2 résume les travaux précédents dans la partie 1, la deuxième partie sera présenté avec l'approche en chapitre 3.

1.4 Conclusion et Bilan

Dans la vie quotidienne, nos décisions sont souvent prises sur la base d'intuitions et d'expériences passées. Elles sont issues d'heuristiques observables et systématiquement, mais en réalité ce type de stratégies ne peut s'appliquer qu'à des problèmes familiers. Lorsque nous sommes confrontés à des situations nouvelles, la tâche de prise de décision devient beaucoup plus difficile. De nos jours, l'environnement des décideurs est de plus en plus complexe et évolue rapidement. La tendance est plutôt à l'accroissement de cette complexité. Cet accroissement est dû à plusieurs facteurs : la technologie de l'information et des ordinateurs, une complexité structurelle des décisions, une plus forte compétition, l'ouverture au marché international, etc. L'utilisation de différents types de SI devient inévitable.

Dans cette partie, nous avons vu l'évolution des SI qui a donné naissance à deux systèmes qui semble similaires mais qui restent différents et complémentaires, que ça soit dans leur façon de manipuler ou gérer des données différentes ou leur degré de prise de décision incomparable où les DSS restent les systèmes principaux pour ce genre de traitement.

Nous avons aussi présenté une étude comparative de ces méthodologies existantes (basée sur un ensemble de critères pertinents comme l'adaptabilité, la fiabilité et l'objectif de décision) et l'identification d'un ensemble de problèmes, notamment l'absence de mécanisme de tolérance aux fautes et l'absence de vérification formelle que nous allons voir en détail dans le chapitre suivant.

Dans le chapitre 3 nous proposons une modélisation de DDSS en utilisant l'approche SMA et on se basant sur les faiblesses des modélisation existantes.

Chapitre 2

Paradigme Skyline

Sommaire

2.1	Introduction	29
2.2	Requêtes Skyline	30
2.2.1	Principe	30
2.2.2	Propriétés de base	31
2.3	Skyline de groupe	32
2.3.1	Définitions	32
2.3.2	Propriétés de base	34
2.4	Calcul du Skyline	35
2.4.1	Travaux sur le Skyline traditionnel	35
2.4.2	Travaux sur le Skyline de groupe	35
2.5	Conclusion et Bilan	39

2.1 Introduction

De nos jours, l'analyse multicritères et la prise de décision deviennent de plus en plus complexes en raison de critères contradictoires et de la complexité des requêtes. L'opérateur Skyline [BKS01a], connu sous le nom de Maxima en géométrie computationnelle ou Pareto dans le domaine de la gestion d'entreprise, gère cette complexité en utilisant la dominance de Pareto. Il extrait des objets intéressants d'un ensemble de données en respectant les préférences de l'utilisateur. L'approche Skyline rencontre notamment un grand succès dans le domaine des bases de données, elle a connu un intérêt exponentiel en raison du bénéfice qu'on peut en tirer, même dans des contextes réels.

Les requêtes Skyline renvoient alors les points les plus intéressants au sens de la relation de dominance de Pareto définie comme suit : soit a et b deux points (ou tuples) avec le même nombre d'attributs (appelés aussi dimensions), a domine b , notez : $(a \prec b)$, si a est aussi bon que b dans toutes les dimensions et meilleur que b dans au moins une dimension. Si ni $a \prec b$ ni $b \prec a$, alors a et b sont incomparables.

De nombreuses propositions et travaux de recherche ont été publiés pour étudier et optimiser la sémantique et le calcul de Skyline, des algorithmes efficaces ont été proposés pour récupérer des objets qui présentent la combinaison optimale des attributs de l'ensemble de données [BKS01a, KRR02, TEO⁺01, CGGL03, GSG⁺05, PTFS03a, PTFS05a].

Récemment, la définition de Skyline s'est avérée médiocre. Considérons un exemple où un utilisateur souhaite obtenir les meilleures équipes de volley-ball d'un ensemble de joueurs, l'approche Skyline traditionnelle renvoie les meilleurs joueurs mais pas les meilleures combinaisons pour créer une équipe qui ne peut être dominée par aucune autre équipe existante générée à partir du même ensemble de joueurs. Un autre exemple similaire consiste à choisir un groupe d'experts pour examiner et évaluer des articles en fonction de la force collective des experts sur les multiples compétences souhaitées. Cela a conduit au concept de Skyline de groupe [IP12a, ZLH⁺13], noté G-SKY, qui est très pertinent et utile dans de nombreux autres domaines comme : la recommandation de groupes, la sélection d'investissements, la détection d'incendie/de crime (lieux les plus dangereux), etc.

Ce nouveau concept a traité dû à leur complexité de nouveaux problèmes pour la communauté Skyline, par exemple, générer des groupes et renvoyer les groupes Skyline les plus intéressants, sont devenus les problèmes les plus difficiles. Certaines solutions à ces problèmes ont été proposées dans la littérature [YZZ⁺16, YZM⁺17]. Cependant, l'interrogation d'un jeu de données à dimension d à l'aide de requêtes Skyline de groupe peut conduire à deux scénarios particuliers : (i) un grand nombre de groupes Skyline renvoyés, ce qui pourrait être moins informatif pour les décideurs, (ii) un nombre réduit de groupes Skyline renvoyés, ce qui pourrait être insuffisant pour les décideurs. Pour résoudre le premier problème (i), diverses approches [ZZLL17b, JZL⁺15, ZLLX19b, JZL⁺15, SCL⁺10] sont proposées pour affiner l'ensemble du Skyline de groupe, réduisant ainsi sa taille, mais aucun des travaux existants n'a abordé le problème (ii) relaxer le Skyline de groupe afin d'augmenter le nombre de résultats de groupes Skyline et ainsi mieux satisfaire les besoins des décideurs. Considérons le problème suivant consistant à trouver les groupes Skyline (où l'indique le nombre d'éléments dans chaque groupe à partir d'un ensemble de données) : un décideur (le formateur) veut obtenir les meilleurs 5 groupes de joueurs. 3 en maximisant les points et le nombre de blocs. Le système renvoie 2 groupes de 3 joueurs de volley chacun. Malheureusement, cette réponse ne satisfait pas le décideur, il/elle a besoin de 5 équipes des meilleurs joueurs mais la définition traditionnelle de Skyline de groupe ne peut renvoyer que 2 groupes. C'est la question principale abordée dans ce chapitre. La solution préconisée vise à élargir la taille de Skyline du groupe en appliquant un processus de relaxation

approprié en utilisant le principe de la logique floue.

Le chapitre 2 est organisé comme suit. Dans la Section 1 et 2 nous introduisant le paradigme Skyline et dans la section 2 nous expliquant le passage vers le Skyline de groupe. La Section 3 est consacrée aux travaux connexes sur le Skyline et sur le groupe. La Section 4 conclut ce chapitre.

2.2 Requêtes Skyline

2.2.1 Principe

L'opérateur Skyline, renvoie des objets parmi un ensemble de données qui fournissent des compromis optimaux par rapport à plusieurs dimensions. Il est basé sur le concept de dominance de Pareto qui peut être défini comme suit :

Soit $\mathbb{D} = (D_1, D_2, \dots, D_d)$ un ensemble de points de données à d dimensions (qui correspond à un ensemble de tuples d'une base de données). On définit une relation $R(A_1, A_2, \dots, A_d)$ dans \mathbb{D} et on suppose l'existence d'une relation d'ordre total sur chaque domaine D_i (domaine d'attribut A_i).

La requête Skyline traditionnelle est basée sur la relation de dominance de Pareto définie comme suit. Soit a et b deux points différents dans \mathbb{D} , Définition 1. a domine (au sens de Pareto) b , noté $a \prec b$, si pour tout i , $a[i] \leq b[i]$, et pour au moins un i , $a[i] < b[i]$, où $a[i]$ (Resp. $b[i]$) est la valeur du point a (Resp. b) pour l'attribut A_i et $1 \leq i \leq d$. Formellement,

$$a \prec b \Leftrightarrow \forall i \in \{1, \dots, d\} : a[i] \leq b[i] \text{ et } \exists j \in \{1, \dots, d\} : a[j] < b[j]$$

Sans perte de généralité, nous considérons dans cette définition que la plus petite valeur est la meilleure.

Le Skyline de \mathbb{D} , noté $SKY(\mathbb{D})$, est un ensemble de points qui ne sont dominés (au sens de Pareto) par aucun autre point de \mathbb{D} . Formellement, on a :

$$SKY(\mathbb{D}) = \{a \in \mathbb{D} \mid \nexists b \in \mathbb{D} : b \prec a\}$$

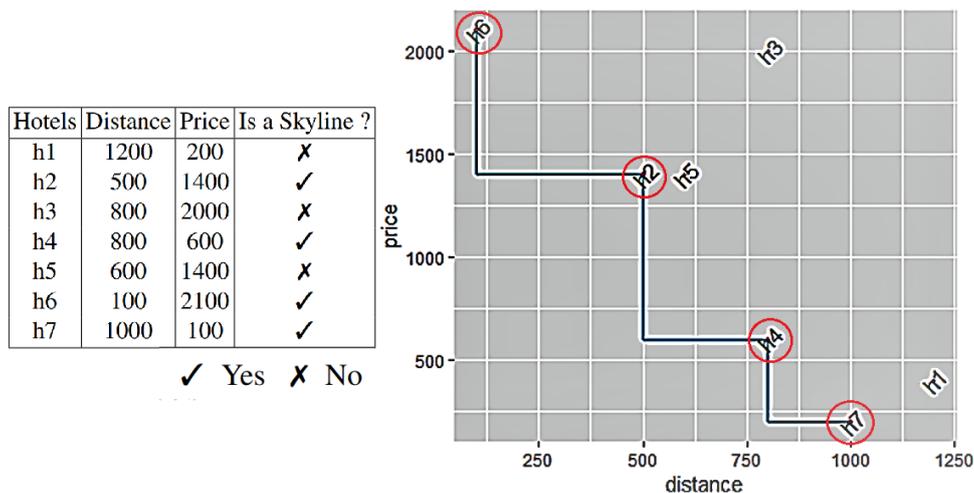


FIGURE 2.1 – Les points Skyline des hôtels de la conférence

Le format de requête skyline SQL¹ est une extension SQL qui intègre une nouvelle clause **SKYLINE** où les préférences de l'utilisateur sont spécifiées [GT07].

La syntaxe SQL skyline proposée basée sur Borzsony [BKS01b] s'écrit comme suit :

“**SELECT * FROM ... WHERE ... SKYLINE ... ORDER BY ...**”

Exemple 1 Prenons l'exemple d'un doctorant participant à une conférence qui souhaite réserver une chambre d'hôtel (voir dans la Figure 2.1 l'ensemble des hôtels). Elle/Il veut que l'hôtel soit proche du lieu de la conférence et aussi qu'il paie un prix raisonnable. La requête SQL Skyline correspondante s'écrit :

“**SELECT** Hotels, Distance, Price **FROM** Hotels **SKYLINE OF** Price MIN, Distance MIN”

où *MIN* spécifie que les deux attributs doivent être minimisés. La figure 2.1 montre les points Skyline encadrés en rouge, nous avons h_3 est dominé par h_2 car la distance et le prix de l'hôtel h_2 sont meilleur que l'hôtel h_3 avec le même principe on obtiendra les points skyline suivants $SKY = \{h_6, h_2, h_4, h_7\}$ qui répondent au mieux à la requête de l'utilisateur pour obtenir les meilleurs hôtels qui répondent aux critères du doctorant.

2.2.2 Propriétés de base

Les requêtes Skyline sont un paradigme populaire et puissant pour incorporer les préférences de l'utilisateur dans les requêtes relationnelles et extraire des points intéressants à partir d'un ensemble de points. La principale différence avec les approches existantes est qu'au lieu de rechercher des vecteurs ou des points, une requête Skyline trouve les maximums sur un ensemble de tuples ou le soi-disant ensemble de tuples Pareto-optimaux. Ces tuples sont ceux qui ne sont dominés par aucun autre tuple dans la même relation. Les meilleures propriétés du Skyline sont :

Définition 1 : Dominance Étant donné des points $p, r \in D$, p domine r , noté $p \prec r$, si et seulement si $\exists j \in [1, d]$ tel que $p.d_j < r.d_j$ et $\forall i \in [1, d] - \{j\} : p.d_i \leq r.d_i$ La dominance a la propriété

d'une relation transitive. C'est-à-dire que si p domine r et r domine t , alors p domine également t . Ceci est donné plus formellement dans la proposition suivante.

Proposition 1 : Transitivité Étant donné les points $p, r, t \in D$, si $p \prec r$ et $r \prec t$, alors $p \prec t$. La transitivité peut être utilisée pour éliminer de toute considération ultérieure un seul point ou un groupe de points qui sont dominés par un point p qui à son tour est dominé par un nouveau point r . Grâce à une analyse précédente, la domination entre deux points a été expliquée. En

contradiction Si les deux points $p, r \in D$ ne se dominant pas (notés $\prec \succ$) simultanément (c'est-à-dire $p \prec \succ r$ et $r \prec \succ p$) sont considérées comme incomparables dans D , et notées $p \sim_D$ ou simplement $p \sim r$. Plus formellement :

Proposition 2 : Incomparabilité

Étant donné deux points $p, r \in D$, si $p \prec \succ r$ et simultanément $r \prec \succ p$, alors p et r sont incomparables sur D (c'est-à-dire $p \sim r$). Cette propriété permet de déterminer si un ou plusieurs points peuvent être des points Skyline. Un point dans l'ensemble des points Skyline doit être incomparable à tous les autres points de l'ensemble.

1. SQL signifie Structured Query Language. SQL nous permet d'accéder et de manipuler des bases de données

2.3 Skyline de groupe

Cette section présente un bref aperçu du Skyline de groupe et de ces notions. Le tableau suivant fournit les symboles avec leurs significations qui seront utilisés dans le reste de ce document.

Symbole	signification
$\mathbb{D}=(D_1, D_2, \dots, D_d)$	<i>un ensemble de points de données à d dimensions</i>
d	<i>Le nombre de dimensions de l'ensemble \mathbb{D}</i>
A_i	<i>l'attribut A_i</i>
D_i	<i>Le domaine de A_i</i>
\mathbf{Q}	<i>Un point de \mathbb{D}</i>
$g_i = (Q_1^i, \dots, Q_l^i)$	<i>Un groupe de l points de \mathbb{D}</i>
$\mathbb{G} = (g_1, g_2, \dots, g_l)$	<i>Ensemble de groupes de taille l de \mathbb{D}</i>
\mathbf{F}	<i>Une fonction d'agrégation</i>
\mathbf{MP}	<i>la relation de préférence (<i>Much preferred</i>)</i>
\mathbf{SKY}	<i>L'ensemble des points Skyline traditionnels</i>
$\mathbf{G-SKY}$	<i>Le skyline de groupe (ensemble de groupes de skyline)</i>
\mathbf{Rest}	<i>L'ensemble des groupes non Skyline</i>
$\mathbf{RG-SKY}$	<i>Le Skyline de groupe relaxé</i>

TABLEAU 2.1 – Symboles et leurs significations

2.3.1 Définitions

De nombreuses applications du monde réel nécessitent de choisir des groupes d'objets, le sport en fait partie, sélectionner la meilleure équipe sur la base d'une liste d'athlètes nécessite de comparer les équipes de joueurs par performance. Un autre exemple consiste à choisir un groupe d'experts pour examiner et évaluer les articles en fonction de la force collective des experts sur les multiples compétences souhaitées. Le concept de Skyline de groupe est également important dans d'autres domaines tels que : la recommandation de groupe, la sélection d'investissements, la détection d'incendie/crime (lieux les plus dangereux), etc.

Ce nouveau type d'extension de requêtes skyline repose sur une relation de dominance entre les groupes, également appelée requêtes skyline combinatoires. Le Skyline de groupe renvoie les groupes qui ne sont dominés par aucun autre groupe dans l'ensemble de données. La relation de dominance entre les groupes peut être formulée de deux manières différentes comme suit :

- La première formulation repose sur la dominance de Pareto et une fonction d'agrégation convenablement choisie. La dominance entre les groupes est nommée *G-Skyline*.
- La seconde sémantique introduit une relation de dominance généralisée particulière appliquée sur des permutations de groupes à comparer. Ce type de dominance est nommé *G-Dominance*.

Ces requêtes conduisent généralement à une grande combinaison de points et une taille de sortie incitant les auteurs à concevoir des algorithmes efficaces pour répondre à ces problèmes. Le Skyline des points de combinaison ou des groupes est définie comme suit (où g et g' sont deux groupes de même taille) [ZZLL17c, LXP⁺15a] :

- **Définition 1 (G-Skyline)**. Soit F une fonction d'agrégation et $g = \{Q_1, Q_2, \dots, Q_l\}$ (resp. $g' = \{Q'_1, Q'_2, \dots, Q'_l\}$) un groupe représenté par un point Q (resp. Q') avec $Q = F(Q_1, Q_2, \dots, Q_l)$ (resp. $Q' = F(Q'_1, Q'_2, \dots, Q'_l)$). Pour deux groupes distincts g et g' , g domine g' (noté $g \prec_{gs} g'$) si Q domine Q' ($Q \prec Q'$) au sens de Pareto.

Nom (Indexé / Non-indexé)	Limites & problèmes
Algo. Index - 2001 (Indexé) Khadidja Berrouigat (2015); Tan et al. (2001)	- Ne prend pas en charge les préférences définies par l'utilisateur (l'ordre des points renvoyés est fixe et dépend de la distribution des valeurs de données)
Algo. Bitmap - 2001 (Indexé) Khadidja Berrouigat (2015); Tan et al. (2001)	- Gestion de mémoire réduite (conversion de points en structure Bitmap) - Gestion inefficace des mises à jour (implique le recalcul de tous les vecteurs de bits) - Non fonctionnel sur plusieurs dimensions - L'utilisateur n'exprime pas ses préférences - L'obligation de coder toutes les valeurs de points
Algo. NN - 2002 (Indexé) Kossmann et al. (2002); Vinh	- Problème de performance, dans le cas simple (la division à 4 régions avant la vérification du nombre de points existants) - inefficace dans les grandes masses de données
Algo. BBS - 2003 (Indexé) Papadias et al. (2003, 2005); Vinh	- Ne fonctionne pas après 5 dimensions
Algo. BNL - 2001 (Non-indexé) Borzsony et al. (2001); Vinh	- Nécessite de nombreuses itérations avant que le Skyline finale soit calculé (analyse toutes les données) - La taille de la fenêtre est limitée, la complexité de l'algorithme dépend de cette taille - Un temps de comparaison non négligeable est nécessaire - Les points Skyline ne sont pas définis progressivement (changent au cours du processus)
Algo. D&C - 2001 (Non-indexé) Borzsony et al. (2001)	- Les points Skyline ne sont pas définis progressivement - Le processus devient inutile sur les données petites (plus efficaces sur une grande quantité de données). - L'algorithme scanne toute la base de données
Algo. SFS - 2005 (Non-indexé) Chomicki et al. (2003)	- Scanne toutes les données - La nécessité de définir une bonne fonction monotone
Algo. LESS - 2005 (Non-indexé) Godfrey et al. (2005)	- Le mécanisme Elimination-filtre (EF) peut devenir complet - Toutes les données doivent être scannées au moins une fois

TABLEAU 2.2 – Comparaison des algorithmes Skyline

Joueurs	points	rebondit
p_1	3	3
p_2	0	4
p_3	4	1
p_4	2	3
p_5	2	2
p_6	2	1

TABLEAU 2.3 – Les données des joueurs

— **Définition 2 (G-Dominance)**. Considérons les deux groupes précédents g et g' , g G-domine g' (noté $g \prec_{gd} g'$), si l'on peut trouver deux permutations de l points pour g et g' , $g = \{Q_{u1}, Q_{u2}, \dots, Q_{ul}\}$ et $g' = \{Q'_{v1}, Q'_{v2}, \dots, Q'_{vl}\}$, tel que $Q_{ui} \preceq Q'_{vi}$ pour tout i ($1 \leq i \leq l$) et $Q_{ui} \prec Q'_{vi}$ pour au moins un i .

Le Skyline de groupe d'un ensemble de données \mathbb{D} , noté $G\text{-Sky}(\mathbb{D})$, est un ensemble de groupes qui ne sont dominés par aucun autre groupe de \mathbb{D} dans le sens de la définition 1 ou 2.

L'ensemble $G\text{-SKY}$ pour représenter la relation de Pareto dominance de groupe peut être défini comme suit :

$$G - SKY(\mathbb{D}) = \{g \in \mathbb{G} \mid \nexists g' \in \mathbb{G} : g' \times g\}$$

où \times représente la relation \prec_{gs} ou \prec_{gd} .

Exemple 2 prenons un exemple consistant en la sélection des joueurs de l'équipe. Supposons que nous ayons six joueurs p_1, \dots, p_6 indiqués dans le tableau 2.3 et que nous ayons besoin de joueurs d'équipe formés de deux joueurs. Dans le tableau 2.4, on génère tous les groupes possibles de

2. $Q_j \preceq Q'_j$ signifie que $Q_j \prec Q'_j$ ou $Q_j \sim Q'_j$ où \sim représente la relation d'indifférence (c'est-à-dire même préférence). La relation d'indifférence se réduit à l'égalité si chaque domaine D_i est doté d'un ordre total.

Groupes	Joueurs	MAX (points,rebondit)	Les groupes Skyline
g_1	$p_1(3,3) - p_2(0,4)$	$G_1(3,4)$	X
g_2	$p_1(3,3) - p_3(4,1)$	$G_2(4,3)$	X
g_3	$p_1(3,3) - p_4(2,3)$	$G_3(3,3)$	X
g_4	$p_1(3,3) - p_5(2,2)$	$G_4(3,3)$	X
g_5	$p_1(3,3) - p_6(2,1)$	$G_5(3,3)$	X
g_6	$p_2(0,4) - p_3(4,1)$	$G_6(4,4)$	✓
g_7	$p_2(0,4) - p_4(2,3)$	$G_7(2,4)$	X
g_8	$p_2(0,4) - p_5(2,2)$	$G_8(2,4)$	X
g_9	$p_2(0,4) - p_6(2,1)$	$G_9(2,4)$	X
g_{10}	$p_3(4,1) - p_4(2,3)$	$G_{10}(4,3)$	X
g_{11}	$p_3(4,1) - p_5(2,2)$	$G_{11}(4,2)$	X
g_{12}	$p_3(4,1) - p_6(2,1)$	$G_{12}(4,1)$	X
g_{13}	$p_4(2,3) - p_5(2,2)$	$G_{13}(2,3)$	X
g_{14}	$p_4(2,3) - p_6(2,1)$	$G_{14}(2,3)$	X
g_{15}	$p_5(2,2) - p_6(2,1)$	$G_{15}(2,2)$	X

✓ Oui X Non

TABLEAU 2.4 – Les équipes Skyline (en utilisant la définition 1)

deux joueurs. Le manager aime extraire les meilleures équipes en fonction des points marqués et des attributs de rebonds des joueurs (e.a. **la plus grande valeur, la meilleure**).

- Basé sur la **Définition 1** et la fonction d'agrégation MAX sur chaque attribut, les valeurs des points et des rebonds de chaque joueur des équipes générées sont donnés dans la colonne 3 du tableau 2.4. Par exemple, g_1 a deux joueurs p_1 et p_2 , les valeurs de $g_1 = (max(3, 0), max(3, 4)) = (3, 4)$. Maintenant, en appliquant la dominance de Pareto sur les équipes générées, on peut vérifier que le skyline du groupe ne contient que l'équipe g_6 .
- Basé sur la **Définition 2**, considérons les deux groupes $g_1 = \{p_1(3,3), p_2(0,4)\}$ et $g_7 = \{p_2(0,4), p_4(2,3)\}$. Puisqu'ici la plus grande valeur est la meilleure, on peut vérifier que g_1 g-domine g_7 . En effet, on peut trouver une permutation pour g_1 donnée par $\{p_2(0,4), p_1(3,3)\}$ telle que $p_2(0,4) \succeq p_2(0,4)$ et $p_1(3,3) \succ p_4(2,3)$ (où \succeq et \succ sont respectivement *préféré ou même préférence*, et *la relation de dominance de Pareto* basées sur les relations d'ordre \geq et $>$). Par conséquent, g_7 n'est pas un Skyline de groupe. Le même processus nous amène à un ensemble de groupes skyline $\{g_1, g_2, g_3, g_6\}$, ce sont les seuls groupes skyline comme aucun autre groupe avec 2 points ne peut g-dominer g_1, g_2, g_3 et g_6 .

2.3.2 Propriétés de base

Dans cette partie, nous présentons quelques propriétés des groupes G-Skyline et des concepts associés qui seront utilisés dans la conception de notre algorithme pour calculer les groupes G-Skyline dans le chapitre 4.

Propriété 1. (Asymétrie). Soit deux groupes G et G de même taille. Si $G \prec_g G$, alors $G \not\prec_g G$.

Propriété 2. (Transitivité). Étant donné trois groupes G_1, G_2 et G_3 de même taille. Si $G_1 \prec_g G_2$ et $G_2 \prec_g G_3$, alors $G_1 \prec_g G_3$.

Lemme 1. Un point dans un groupe G-Skyline ne peut pas être dominé par un point extérieur au groupe.

Preuve. Par contradiction, supposons qu'un point p_i dans un groupe G-Skyline G est dominé par un point p_j extérieur au groupe. Si nous utilisons p_j pour remplacer p_i dans G , le nouveau groupe dominera G puisque p_j domine p_i et tous les autres points sont les mêmes, ce qui contredit la définition de G-Skyline.

2.4 Calcul du Skyline

Nous passons en revue les travaux, les plus importants, de calcul du G-SKY. Mais tout d'abord, rappelons les principaux algorithmes de calcul du SKYline (SKY) en mettant l'accent sur les limites de chaque algorithme.

2.4.1 Travaux sur le Skyline traditionnel

Le calcul du skyline a été introduit au début comme le calcul des maxima d'un ensemble de vecteurs par Kung et al. [KLP75] en géométrie computationnelle. Plus tard, de nombreux travaux ont été effectués pour développer des algorithmes plus efficaces. Ces derniers s'inspirent des deux méthodes existantes qui peuvent être classés en deux catégories pour le calcul des points Skyline : Les algorithmes séquentiels ou non indexé, qui s'appuie sur des index précalculés sur les données. Les méthodes indicielles ont de meilleures performances, car elles évitent d'accéder à l'ensemble de la collecte de données, mais ils ont une applicabilité limitée en raison de la nécessité d'un jeu de données indexé qui peut être exponentielle. De plus, les index multidimensionnels comme les R-trees ont leurs propres limites car ils souffrent de la malédiction bien connue de la dimensionnalité. Les méthodes non indexées sont plus génériques, en ce sens qu'ils ne nécessitent aucune structure d'accès spécialisée pour calculer les points Skyline.

Après l'étude des différents algorithmes Skyline, le tableau 2.2 présente les limites et les problèmes de chaque algorithme Skyline existant.

2.4.2 Travaux sur le Skyline de groupe

Le Skyline de groupe a été introduit pour traiter de nouveaux besoins de certaines applications modernes du monde réel. En effet, la décision à prendre requiert le choix d'un groupe d'objets plutôt que des objets individuels parmi un ensemble de données.

Nous passons ici en revue les principaux travaux de recherche liés au skyline de groupe tant de point de vue calculatoire que sémantique. Nous fournissons également une comparaison complète de ces travaux par rapport à un ensemble de critères convenablement choisis.

La méthode brute pour obtenir G-SKY consiste à énumérer tous les groupes (n : tuples, k : taille du groupe), puis à exécuter la requête en fonction de la relation de dominance du groupe. La méthode de la force brute calcule le tuple agrégé pour chaque groupe, puis utilise n'importe quel algorithme de Skyline traditionnel pour trouver le Skyline de groupe. Cette méthode prend beaucoup de temps et le coût de stockage peut être exponentiel en raison de l'énorme entrée intermédiaire pour l'algorithme de tuple du Skyline traditionnel. Dans [ZLH⁺14] quelques alternatives à cette méthode naïve sont proposées. Les propositions de Skyline de groupe existantes se concentrent sur le problème (i) en utilisant l'une des définitions présentées dans la section 2.3 sur le flux où les données statiques, par exemple. Les articles [JZL⁺15, SCL⁺10, YZM⁺17, YZZ⁺16, ZGHW18, ZLLX19a, ZZLL17c] combinent les avantages des requêtes skyline de groupe et le principe du top-k meilleurs réponses. Un autre travail [LXP⁺15b] présente une structure qui repré-

sente les points dans un graphe Skyline orienté et capture toute la relation de dominance entre les points basée sur la notion de couches Skyline. Certains articles [DLCL18a, GLW⁺16a, LYX⁺18] se concentrent sur les flux de données pour calculer G-SKY en continu, où ils invoquent le problème du calcul de G-SKY lorsqu'un nouveau point p arrive dynamiquement. L'idée sous-jacente est de stocker des informations de dominance qui pourraient être réutilisées dans un autre espace de recherche. Dans [IP12b, LLH17], les auteurs génèrent des groupes de candidats de manière progressive et mettent à jour dynamiquement le Skyline de groupe résultant. D'autres articles [CSL13, CD18, GXI12, YLP⁺19, YQL⁺17, ZLYL18, ZLLZ19, ZZLL17a, ZZL⁺17] se concentrent sur l'optimisation des performances expérimentales de l'algorithme Skyline de groupe en utilisant la parallélisation et d'autres méthodes. Sur la base de nos précédents articles [NHS18, NSH18], nous résumons et comparons les travaux existants ci-dessus dans le tableau 2.5 (où (i) **D.type (type de données)** : désigne flux de données ou données statiques, (ii) **Perf (Performance)** : signifie que le travail se concentre sur le temps d'exécution plutôt que sur la qualité des réponses, (iii) **Def (Definition)** : indique quelle définition 1 ou 2 utilisé pour calculer le groupe skyline, (iv) **Ref (Raffinement)** : signifie que l'étude est dotée d'un processus de raffinement pour réduire l'ensemble de groupes skyline renvoyés et (v) **Rel (Relaxation)** : signifie que le travail essaie d'obtenir plus de groupes de Skyline en assouplissant la définition du Skyline de groupe traditionnel).

Les travaux existants peuvent être classés en deux parties (selon les deux définitions précédentes) et sont utilisés sur les données statiques ou dynamique (steam), Nous expliquons le fonctionnement des travaux les plus importants :

- a) **Requêtes Skyline de Groupe et Top-k.** Les travaux dans (Zhu, 2017) [ZZLL17b] renvoient k groupes Skyline qui dominent le nombre maximal de points dans un ensemble de données donné. Ils combinent les avantages de groupe Skyline et des requêtes top-k. Il s'agit de premiers travaux qui abordent le problème de trouver les top-k groupes Skyline en utilisant un vecteur Bit pour stocker le nombre dominé de chaque point (mis à 0 si le point est dominé, à 1 sinon).
- b) **Groupes optimaux de Pareto.** (Liu, 2015) [LXP⁺15a] ils proposent un algorithme dédié au calcul des k premières couches Skyline et au calcul des groupes G-Skyline à l'aide de deux propositions : l'algorithme point par point (l'idée est d'étendre les groupes possibles sur une liste ordonnée de points) et l'algorithme par groupe d'unités (développe les groupes candidats un point à la fois). Ils présentent une structure qui représente les points dans une direction graphique Skyline et capture toutes les relations de dominance entre les points sur la base de la notion de couches Skyline. Cette structure utilise un graphique Skyline orienté où un nœud représente un point et une arête représente une relation de dominance.
- c) **Points Skyline basés sur un groupe dans le contexte de réseau de capteurs avec des données flux.** (Dong, 2018) [DLCL18b], ce travail se concentre sur les flux de données, où ils invoquent le problème du calcul de G-Skyline lorsqu'un nouveau point p arrive. Tout d'abord, ils vérifient à quelle couche appartient le point p , puis mettent à jour le graphique pour construire les nouvelles relations entre tous les points, enfin, ils calculent le G-Skyline en fonction de la stratégie de partage. Afin de calculer le G-Skyline en continu, ils calculent les couches Skyline et le graphique pour tous les points actifs plutôt que les k premières couches Skyline et le graphique (le même processus est utilisé lorsque p est retiré).
- d) **Groupes Skyline.** (Li, 2012) [LZH⁺12] ils ont identifié deux propriétés anti-monotones avec des degrés d'applicabilité variables. Une propriété spécifique à l'ordre ainsi qu'une propriété de génération de candidat faible. Les auteurs proposent trois techniques, à savoir la compression de sortie, l'élagage d'entrée et l'élagage de l'espace de recherche. Pour les

Travaux existants	D.type	Perf.	Def.	Ref.	Rel.
<i>Efficient computation of combinatorial skyline queries</i> [CSL13]	Statique	✓	1	✓	✗
<i>An Efficient Algorithm to Compute Compositional Skyline</i> [CD18]	Statique	✓	Aucune	✓	✗
<i>Finding Group-Based Skyline over a Data Stream in the Sensor Network</i> [DLCL18a]	Dynamique (Stream)	✓	2	✓	✗
<i>Efficient processing of skyline group queries over a data stream</i> [GLW ⁺ 16a]	Dynamique	✓	2	✓	✗
<i>Combination skyline queries</i> [GXI12]	Statique	✓	1	✓	✗
<i>Group skyline computation</i> [IP12b]	Statique	✓	2	✓	✗
<i>Incremental evaluation of top-k combinatorial metric skyline query</i> [JZL ⁺ 15]	Statique	✓	Aucune	✓	✗
<i>Progressive approaches for Pareto optimal groups computation</i> [LYX ⁺ 18]	Dynamique	✓	Aucune	✓	✗
<i>Discovering Group Skylines with Constraints by Early Candidate Pruning</i> [LLH17]	Statique	✓	1	✓	✗
<i>Finding pareto optimal groups : Group-based skyline</i> [LXP ⁺ 15b]	Statique	✓	1	✓	✗
<i>Top-k combinatorial skyline queries</i> [SCL ⁺ 10]	Statique	✓	1	✓	✗
<i>Identifying Most Preferential Skyline Product Combinations</i> [YZM ⁺ 17]	Statique	✓	Aucune	✓	✗
<i>Identifying most preferential skyline product combinations under price promotion</i> [YZZ ⁺ 16]	Statique	✓	None	✓	✗
<i>Efficient Contour Computation of Group-based Skyline</i> [YLP ⁺ 19]	Statique	✓	2	✓	✗
<i>Fast algorithms for pareto optimal group-based skyline</i> [YQL ⁺ 17]	Statique	✓	2	✓	✗
<i>Finding k-Dominant G-Skyline Groups on High Dimensional Data</i> [ZGHW18]	Statique	✓	Aucune	✓	✗
<i>On skyline groups</i> [ZLH ⁺ 14]	Statique	✓	2	✓	✗
<i>Finding optimal skyline product combinations under price promotion</i> [ZLYL18]	Statique	✓	Aucune	✓	✗
<i>Top-k Dominating Queries on Skyline Groups</i> [ZLLX19a]	Statique	✓	1,2	✓	✗
<i>Computing skyline groups : an experimental evaluation</i> [ZLLZ19]	Statique	✓	1,2	✓	✗
<i>Computing Skyline Groups : An Experimental Evaluation</i> [ZZLL17a]	Statique	✓	1,2	✓	✗
<i>Top-k Skyline Groups Queries</i> [ZZLL17c]	Statique	✓	1	✓	✗
<i>Parallelization of group-based skyline computation for multi-core processors</i> [ZZL ⁺ 17]	Statique	✓	2	✓	✗

✓ Oui ✗ Non

TABLEAU 2.5 – Comparaison des travaux liés au Skyline de groupe

agrégats MAX et MIN, ils observent que de nombreux groupes peuvent partager le même tuple agrégé. La deuxième idée était l'élagage de l'espace de recherche. Au lieu d'énumérer chaque combinaison de k-uplets, ils excluent de cette façon un grand nombre de combinaisons.

- e) **Skyline de groupe.** (Im, 2012) [IP12a] ils ont proposé GDynamic un équivalent à un algorithme dynamique qui remplit une table de groupes Skyline. Il génère des groupes candidats de manière progressive et met à jour dynamiquement le skyline de groupe résultant. Puisqu'ils génèrent des groupes candidats de manière progressive, la vitesse d'un algorithme Skyline de groupe dépend de l'ordre dans lequel il examine les groupes candidats.
- f) **Groupe Skyline sur des flux de données.** (Guo, 2016) [GLW⁺16b], l'idée de base est de stocker des informations de dominance qui pourraient être réutilisées. Ils ont proposé un algorithme efficace pour les calculer de manière incrémentale. Pour chaque objet actif p, ils maintiennent (1) le nombre de dominants, noté p : num et (2) les objets pouvant être dominés par p, notés p : dominé. Lorsqu'un objet est ajouté ou supprimé, ils mettent à jour p : num et p : dominee de chaque objet influencé par p. Les objets ayant moins de k dominants sont signalés comme candidats. Ainsi, ils peuvent enregistrer les contrôles de dominance des objets actifs qui précèdent

La figure suivante résume les différents travaux connexes en utilisant la définition et les données utilisées.

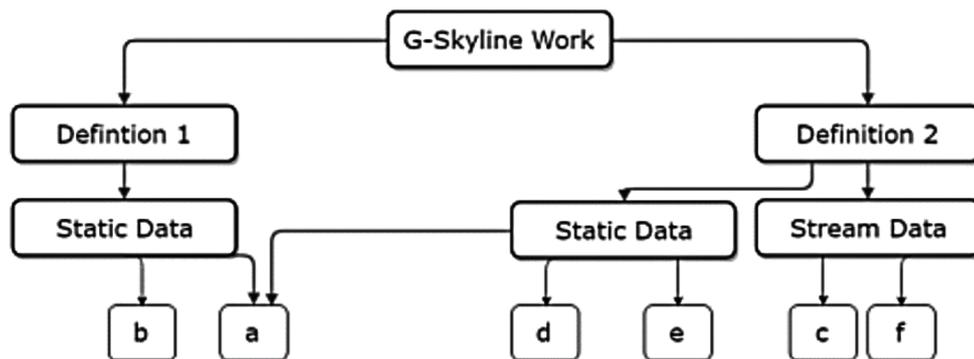


FIGURE 2.2 – Travaux connexes dans le contexte de calcul du Skyline de groupe

2.5 Conclusion et Bilan

Dans cette partie, nous avons présenté la relation de dominance pareto individuelle et de groupe. Nous avons aussi introduit les notions de logique floue qui seront utilisés dans la partie contribution.

Nous avons également présenté et passé en revue les travaux sur le Skyline de groupe. Ce nouveau concept est toujours d'actualité et de nombreux défis sont à prendre en compte, on peut citer le problème de taille de groupe différente, les groupes ne sont pas extraits au fur et à mesure, le grand nombre de combinaisons de points possibles et le grand nombre de groupes de sortie. Certains de ces problèmes peuvent être résolus à l'aide de méthodes de relaxation et d'algorithmes progressifs utilisés dans les algorithmes Skyline individuels précédents. Un calcul parallèle peut également être bénéfique pour ce type de processus utilisant un environnement distribué. Enfin, nous pensons revoir la relation de dominance pour gérer différemment les différentes tailles de groupe et la définition de Skyline de groupe.

Ils y'a plusieurs algorithmes de calcul du Skyline que nous avons présenté dans le tableau 2.2, après une étude comparative de ces algorithmes nous avons choisi le plus optimale et qui le plus adapté à notre contexte de travail, l'algorithme Salsa sera l'algorithme utilisé au chapitre 3 et au chapitre 4.

Nous avons aussi présenté les algorithmes sur le Skyline de groupe pour traiter de nouvelles applications du le monde réel et répondre à différentes requêtes qui nécessitent de choisir un groupe d'objets plutôt que des objets individuels d'un ensemble de données.

Deuxième partie

Contributions

Chapitre 3

Approche SMA pour la modélisation des DDSS

Sommaire

3.1	Introduction	45
3.2	Motivation	46
3.3	Aperçu du DDSS proposé	47
3.3.1	Structure générale de ADS^2	47
3.3.2	Structure interne de ADS^2	48
3.4	Modélisation du comportement du DDSS	49
3.5	Caractéristiques de l'approche ADS^2	51
3.6	Meta-model du framework ADS^2	51
3.7	Intégration du paradigme Skyline dans la prise de décision	52
3.8	Conclusion et Bilan	61

3.1 Introduction

Les DDSS peuvent être représentés et modélisés par des systèmes multi-agents car ils sont particulièrement appropriés pour les systèmes distribués [NA14, BdNdL16]. Les multi-agents permettent de représenter les différents acteurs concernés par la décision. C'est le modèle le plus adéquat pour modéliser des systèmes complexes à partir des entités qui les composent et de leurs interactions en leur donnant un niveau d'abstraction très élevé qui permet d'exprimer l'ensemble des DSS en termes d'agents qui jouent des rôles autonomes et souvent hétérogènes.

L'approche SMA que nous proposons permet la modélisation à base d'agents des systèmes DDSS afin de remédier à un ensemble de problèmes détectés dans les travaux précédents. Nous proposons le modèle *ADS*², une architecture multi-agent basée sur les modules DSS standard, tels que : le décideur, l'interface utilisateur, le moteur de connaissances, la base de données système et le module de traitement. L'*ADS*² est raffinée ensuite par un autre sous-modèle basé sur le Skyline afin de renforcer la prise de décision interne du système globale.

Les processus DSS actuels souffrent de plusieurs aspects, tels que la difficulté de générer des choix utiles à temps et surtout fiables pour les décideurs. Dans la deuxième partie, nous proposons une nouvelle approche pour améliorer le processus de décision basée sur les architectures Salsa (Sort and Limit Skyline algorithm) et HDMSM (Hybrid Decision Making Support Model) pour l'extraction de critères afin d'aider les décideurs à choisir les alternatives les plus appropriées dans un délai acceptable.

Le reste de ce chapitre est organisé comme suit. Dans la Section 1, nous introduisons la problématique et la motivation de modélisation des SMA avec DDSS. Les Sections 2 et 3 sont consacrées à la structure et au comportement du système DDSS proposé. La section 4 présente les caractéristiques de l'approche et le meta-Modèle du système *ADS*². La Section 5 concerne la deuxième contribution pour améliorer le processus interne du DSS. La Section 6 conclut ce chapitre.

3.2 Motivation

Au cours de la dernière décennie, les systèmes d'aide à la décision distribuée (DDSS) ont fait l'objet d'une attention particulière. Un certain nombre d'études ont été proposées tant au niveau de la modélisation qu'au niveau algorithmique. Dans cette section, nous avons sélectionné quelques travaux qui se sont concentrés sur la modélisation du DDSS à l'aide du SMA. Ces propositions peuvent être classées en deux catégories : génériques et spécifiques.

La première ne dépend pas du domaine d'application, voir les travaux de [TAZ09b] (a), [MBA16b] (b), [ANOM12b] (c), [DS04a] (d).

La deuxième catégorie d'approches est appliquée dans un domaine d'étude spécifique, par exemple, le marketing [ASL04b] (e), le diagnostic et le pronostic des tumeurs cérébrales [GVMA⁺06b, LAEM⁺07] (f), les événements sportif [Yin14b] (g).

L'étude de ces approches récentes, de conception et de mise en œuvre de systèmes décisionnels en environnement distribué utilisant des architectures MAS, nous a conduit à extraire et expliciter un ensemble de critères de comparaison obtenus à partir de la conception d'architecture MAS, de l'évaluation des propriétés du DSS et du comportement des systèmes distribués (par exemple, l'adaptabilité de l'environnement, le mode d'apprentissage, l'objectif de décision (qualité/temps), la fiabilité du système, le domaine d'application, etc.).

Nous classons ainsi les différents travaux de recherche concernant ces critères. Le tableau 3.1 résume cette comparaison.

Nous avons choisi de comparer notre proposition avec les autres travaux sur les mêmes paramètres système. La partie II, permet de voir clairement que l'approche *ADS*² est meilleur par rapport aux approches sur plusieurs paramètres, prenant l'exemple des deux paramètres les moins pris en charge par les travaux précédents, la fiabilité du système et la tolérance aux fautes, les agents dynamique injectés et la distribution de l'approche *ADS*² de notre système permet de renforcer ces deux paramètres en ayant un système très renforcé.

Les différents travaux discutés sont théoriquement puissants, mais souffrent encore de certains aspects, les problèmes les plus communément rencontrés par ces approches sont :

- Les DSS internes ne sont pas spécifiés dans la plupart des propositions (processus abstrait).

		Distributed Decision Support Systems Criteria											
Approaches		1	2	3	4	5	6	7	8	9	10	11	12
Part I	a	No	Quality	Is not defined	No	Yes	No	Yes	Is not defined	No (referential)	General	Yes	Yes
	b	No	Quality	Is not defined	No	Yes	No	Yes	Is not defined	No (referential)	General	No	Yes
	c	No	Quality (with a limited time)	Is not defined	No	Yes	No	Yes	Is not defined	No (referential)	General	No	Yes
	d	Yes	Time	Is not defined	Yes	Yes	No	Yes	Is not defined	No	General	Yes	Yes
	e	Yes	Quality	Is defined	No	Yes	No	Yes	Is not defined	Yes	Specific (marketing)	No	No
	f	Yes	Quality	Is defined	Yes	Yes	Yes	Yes	Is not defined	No	Specific (brain tumour)	Yes	Yes
	g	Yes	Quality	Is not defined	Yes	Yes	No	Yes	Is not defined	No	Specific (sport event)	No	No
Part II	ADS ²	YES	Quality & Time	Defined	YES	YES	YES	YES	In process	YES	General	YES	In process

1. Adaptability, 2. Decision objective (quality/time), 3. Internal DSS model, 4. predictability, 5. Learning, 6. Reliability, 7. Interoperability, 8. Formal architecture, 9. Distributed data, 10. Domain (general/specific), 11. Fault tolerance, 12. Implementation

TABEAU 3.1 – Comparaison des travaux connexes avec notre travail

- Les systèmes ne sont pas tolérants aux pannes puisque leur contrôleur (animateur/coordonateur) est centralisé.
- Le type d'agent utilisé est insuffisant et peu sûr (en général agents cognitifs ou architecture de couches).
- Manque d'adaptabilité (auto-organisation) et de dynamisme en fonction des changements
- Le mode d'apprentissage utilisé reste incomplet (raisonnement basé sur des cas similaires).
- Pas de prédictions, mais juste des évaluations de solutions.
- Les décisions ne sont pas exprimées en langage naturel.
- Le temps n'est pas pris en compte dans la prise de décision (sauf pour [DS04a]).

Nous avons également remarqué qu'il n'y a pas de travail proposant une suite d'outils bien définis pour que les concepteurs puissent concevoir des DDSS spécifiques et adaptés à leurs besoins.

Pour combler ces insuffisances, nous proposons *ADS²* (**A**gents **A**pproach for **D**istributed **D**ecision **S**upport **S**ystems), une architecture multi-agent basée sur les modules DSS standard, tels que : le décideur, l'interface utilisateur graphique, le moteur de connaissances, le système de gestion de base de données, le module de traitement [SWC⁺02, KT99].

3.3 Aperçu du DDSS proposé

3.3.1 Structure générale de *ADS²*

La Figure suivante donne un aperçu complet de l'architecture générale que nous avons définie. Le composant principal est l'agent dynamique appelé "Agent principal" qui est l'entité qui démarre le processus en notifiant les différents DSS. Les différents participants commencent leur processus et échangent des données et des alternatives (décisions). Cette architecture s'adapte principalement aux changements externes en introduisant un nouvel agent interne appelé "observateur". Ce dernier surveille le contexte externe. Ensuite, l'observateur informe les autres agents concernés en cas de détection de changements qui peuvent influencer la fiabilité de la décision.

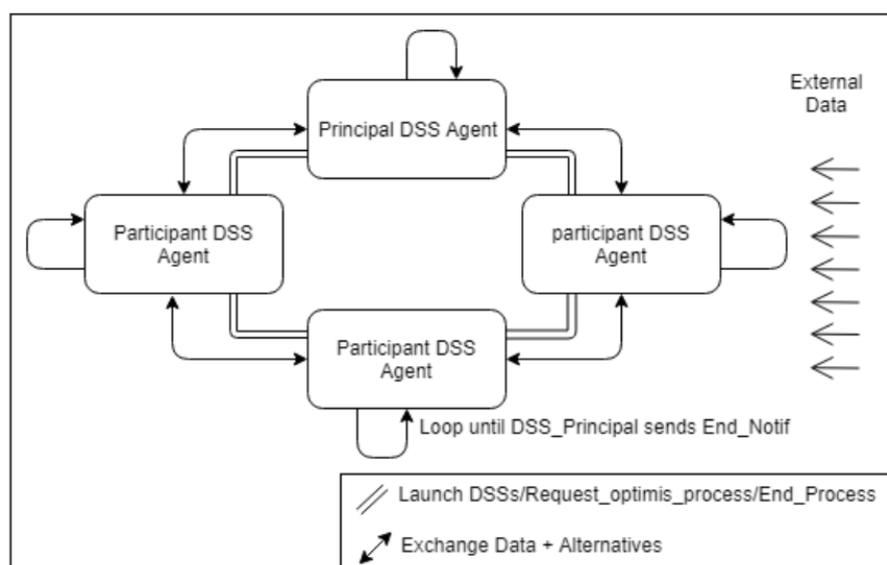


FIGURE 3.1 – Architecture générale de *ADS²*

3.3.2 Structure interne de ADS²

Nous définissons deux types de DSS dans l'architecture externe : Principal et participant. Le DDS principal a la même architecture en tant que participant DSS, mais il contient 2 composants supplémentaires (agent de comparaison et observateur). Il a principalement 6 types d'agents, comme il est décrit ci-dessous.

1. Type d'agent proxy : créé pour une raison spécifique (objectif) puis il sera directement détruit par la suite (implicitement ou explicitement).
2. Type d'agent mobile : est capable de migrer de manière autonome d'un module à un autre à travers le réseau. Une fois le processus terminé, il revient au composant d'origine pour apporter des résultats [DN05, Rod11].
3. Type d'agent miroir : copie l'état d'un autre agent [RCBV15].
4. Type d'agent cognitif : il utilise une représentation symbolique de son contexte d'exécution (base de connaissances) [Fer97a, Lai02].
5. Type d'agent réactif : il ne prend pas en compte le passé et utilise des mécanismes de réaction pour les événements [Fer97a, Lai02].
6. Agent de type BDI : "Croyances, Désirs, Intentions", est une architecture basée sur des extensions logiques, afin de prendre une décision, un agent traitera les étapes suivantes [Fer97a, Lai02, RUF⁺], commençant par une mise à jour des croyances à partir des perceptions. En effet, (i) l'agent définit ces croyances en termes d'information et connaissance de l'environnement actuel. (ii) l'agent a également définit ses désirs en exprimant des états qu'il veut atteindre. (iii) l'agent définit ses intentions en choisissant certains états pour atteindre, le processus se termine par une fonction pour sélectionner l'action à exécuter.

La conception structurelle de l'ADS² est basée sur le processus standard DSS [SWC⁺02, KT99], caractérisé par les étapes suivantes :

- Intelligence : rechercher les conditions qui nécessitent une décision.
- Conception : développer et analyse l'alternative possible actions (solutions).
- Choix : choisir un plan d'action parmi eux.
- Mise en œuvre : adopter l'ensemble des actions choisies dans la situation de décision.

En utilisant le processus précédent, nous proposons la structure interne appelé IDS-MAS (Intelligent Decision Support Multi-Agent System) illustré à la Figure 3.2, il contient 8 agents principaux (Software et Humain) que nous décrivons ci-après.

Agents logiciels : pour ce type, nous avons 6 agents classés en utilisant les types définis précédemment, comme suit :

1. **Model Management Agent : (type : BDI Agent)** le cœur de l'Agent DSS, il démarre le processus de décision en utilisant la requête du décideur et les données courant interne et externes.
2. **Agent d'interface : (type : Agent réactif)** est responsable pour formuler le problème (formulation de requête avec critères) et en interaction avec le décideur.
3. **Data Management Agent : (type : Cognitive Agent)** encapsule le MIS (Management Information System) pour la gestion de la base de données DSS.
4. **Agent de gestion des connaissances : (type : Cognitif Agent)** extrait et crée une connaissance utile de données, l'agent dispose d'un module Machine Learning pour affiner le processus de décision.
5. **Agent observateur : (type : Agent proxy)** devient externe change et calcule le degré, s'il est élevé, l'agent averti les autres agents (et le système doit s'adapter à ce nouveau changement). Cet agent est créé au début du processus.

6. **Comparaison Agent : (type : Proxy Agent)** reçoit le DSS alternatives et calcule le degré d'accord, selon ce degré, l'agent peut demander une nouvelle itération ou arrêter le processus (en respectant le temps T défini par le décideur). Cet agent est créé au début du processus

Agents humains : pour ce type, nous avons 2 agents :

1. **Agent décideur** : génère le problème et les différents critères.
2. **Agent expert** : crée et met à jour les connaissances initiales et tables de décision.

Tous les agents de calcul ont un miroir et un mobile agents (représentés par des cercles sur la Figure suivante) comme étant hautement fautifs tolérant (dans le système interne), une autre possibilité est le code migration si le module de trous ne peut plus fonctionner.

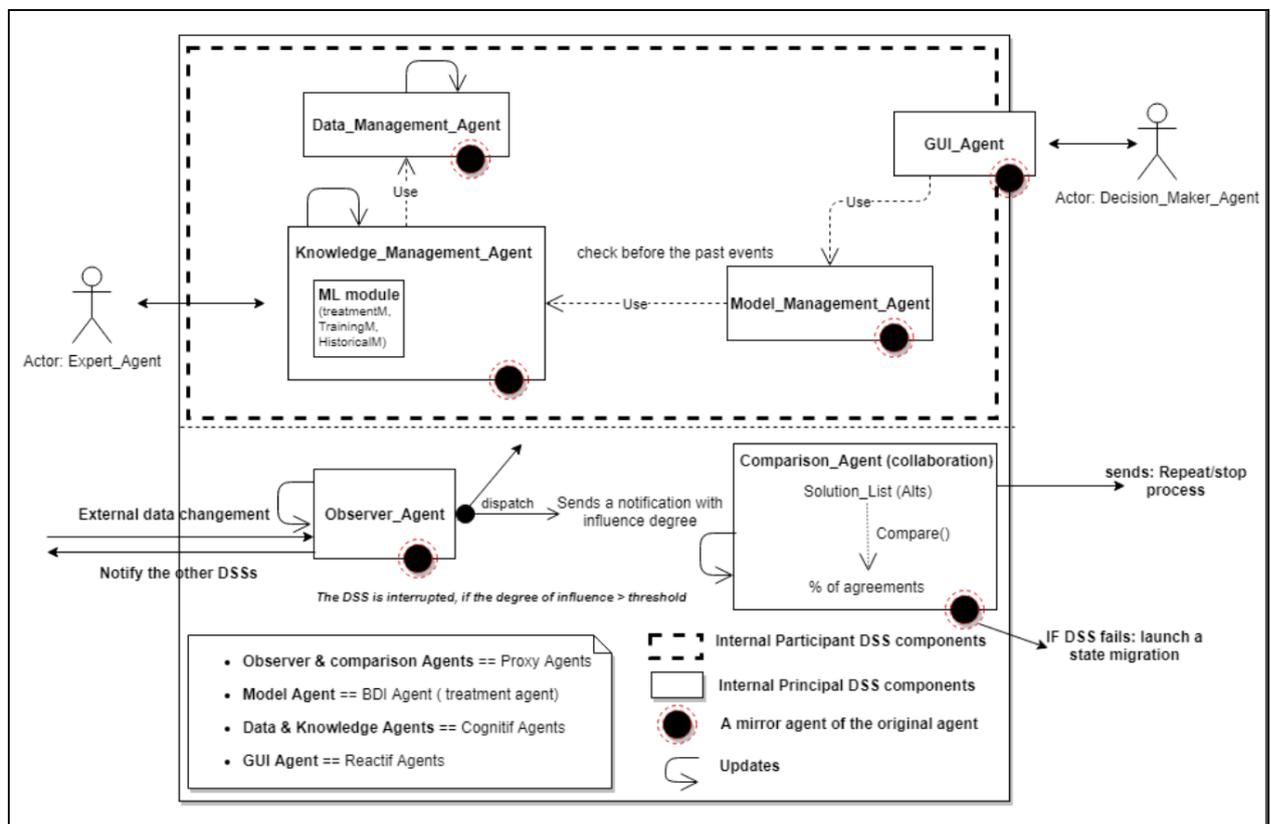


FIGURE 3.2 – Structure interne d'un DSS principal (IDS-MAS)

3.4 Modélisation du comportement du DDSS

En plus de l'aspect structurel de notre proposition, nous avons défini l'aspect comportemental de l'architecture. En fait, chaque agent d'*ADS*² est caractérisé par un comportement spécifique. Ensuite, l'assemblage de ces comportements conduit à un comportement global caractérisant l'ensemble de l'architecture. Nous nous concentrons sur le comportement global en modélisant l'interaction entre les agents à l'aide du diagramme de séquence UML tel qu'il est résumé dans la Figure 3.3.

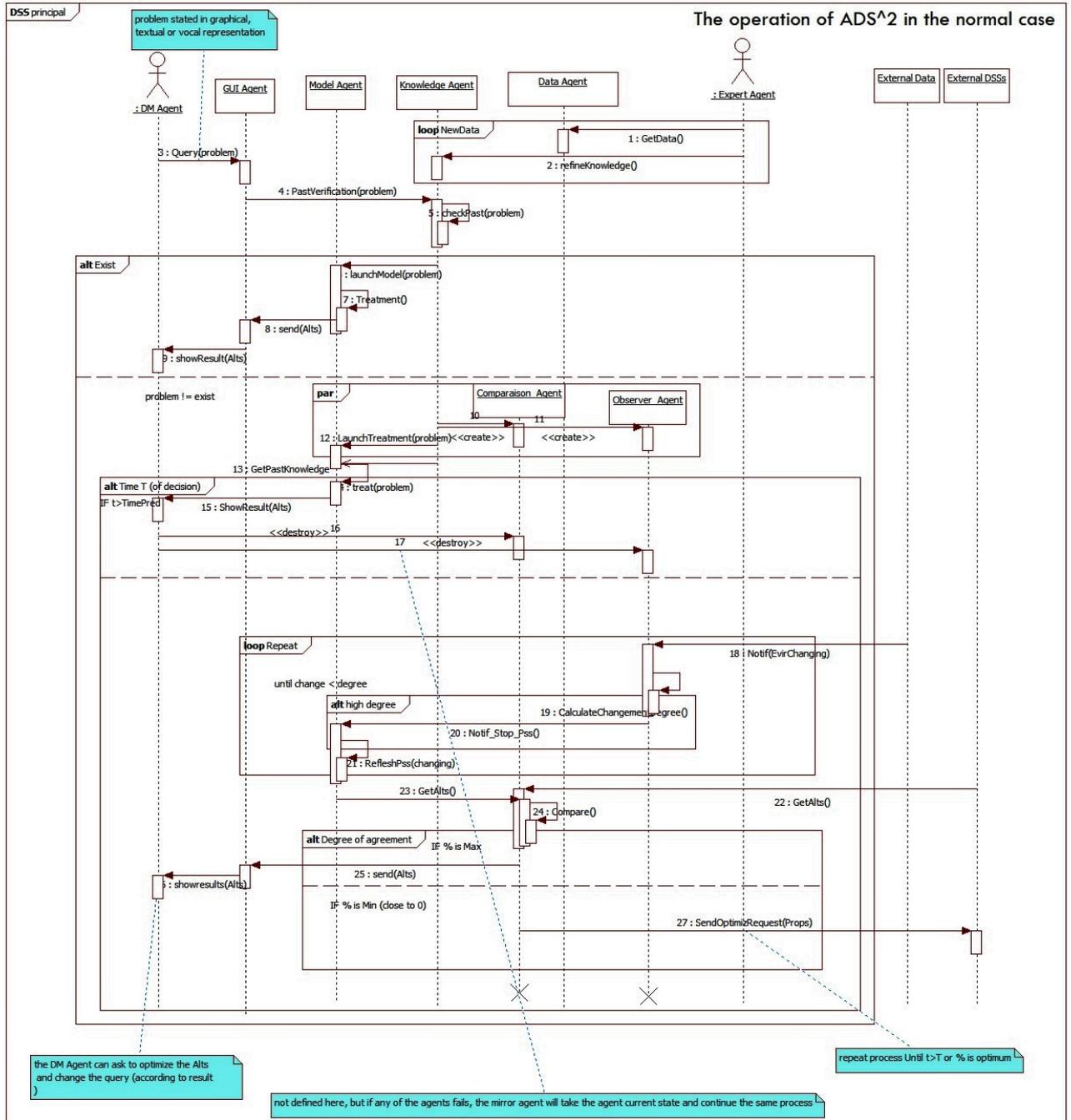


FIGURE 3.3 – Comportement général du système proposé (ADS²)

3.5 Caractéristiques de l'approche ADS²

La spécificité de notre proposition est sa généralité, son dynamisme et l'adaptabilité aux changements rapides de l'environnement avec l'agent d'observation. Le système proposé est capable de prédire ou détecter les pannes dans les plus brefs délais et mettre en œuvre des actions de récupération à l'aide d'agents miroirs. Un agent miroir enregistre l'état actuel (périodiquement), il peut par conséquent remplacer l'agent d'origine et poursuit le processus en cours. En ce qui concerne les critères établis précédemment dans le tableau 3.1 (PARTIE I), nous souhaitons parvenir à un compromis entre le temps de réponse et la qualité de décision. De plus, nous nous concentrons sur les aspects de performance, en ayant un temps limite défini par le décideur au début du processus de prise de décision. Ceci peut être optimisé à l'aide d'agents proxy (créés si nécessaire et détruit juste après avoir obtenu les résultats).

Une autre spécificité de l'ADS² est le mode d'apprentissage. C'est-à-dire que la vérification est faite avant de lancer un nouveau calcul de décision, le système recherche un problème similaire (traité) au problème considéré en utilisant le Machine Learning (qui est un mécanisme interne de l'agent de gestion des connaissances). L'objectif derrière notre approche est d'améliorer la performance qualité et de faciliter le développement de DDSS en aidant les concepteurs depuis les phases de conception et de vérification jusqu'au code phases de génération et de mise en œuvre.

Nous pouvons comparer notre architecture méthodologique ADS² à d'autres en respectant les critères précédents. Tableau 3.1 (PARTIE II) présente les objectifs atteints grâce à l'architecture de ADS².

3.6 Meta-modèle du framework ADS²

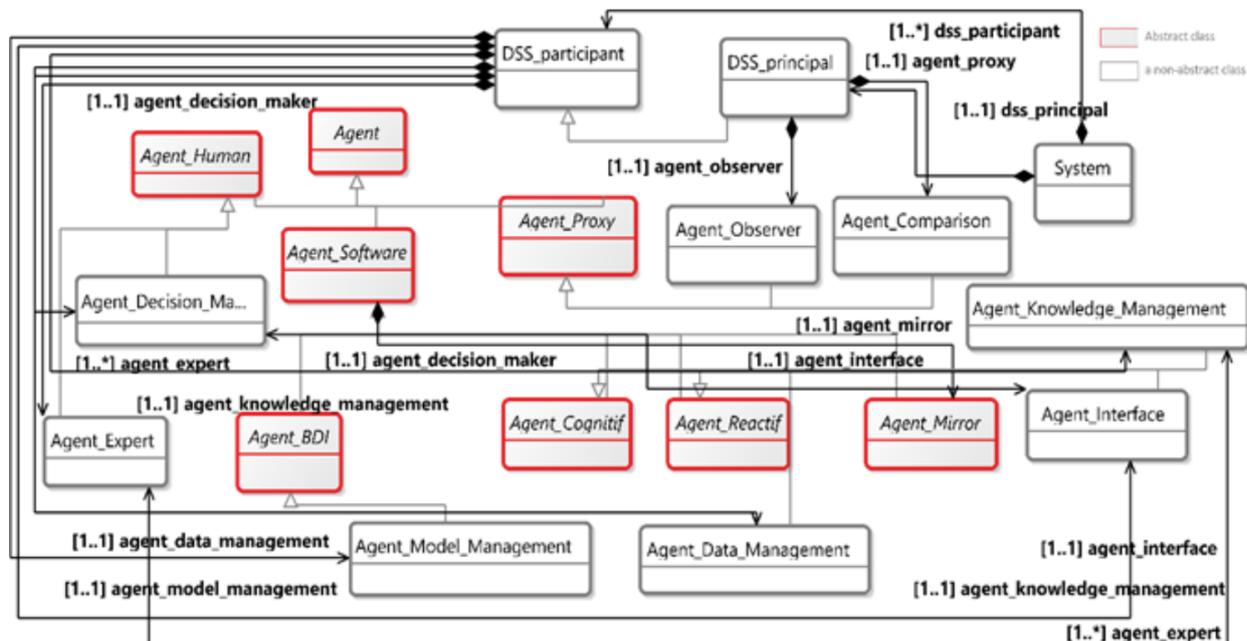
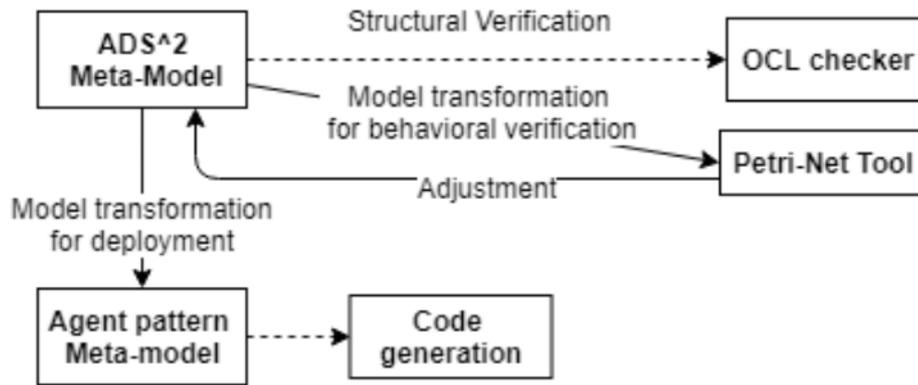


FIGURE 3.4 – Méta modèle de ADS²

Notre objectif est de fournir aux concepteurs des artefacts de modélisation qui les aideront à esquisser leurs besoins et à obtenir un modèle conforme à la méthodologie ADS². En effet,


 FIGURE 3.5 – le Framework de vérification de ADS^2

le but de cette modélisation des artefacts consiste à figer non seulement l'architecture d'un système donné mais aussi la sémantique de ses agents. En conséquence, nous proposons un méta-modèle qui joue le rôle de ADS^2 syntaxe abstraite. La Figure 3.5 montre le méta-modèle exprimé en Ecore (qui est un langage de méta-modélisation implémentant les concepts de Meta-Object Facility [MOF] dans Eclipse IDE [EMF]). Il contient les classes, la composition et les relations d'héritage correspondant aux différents agents qui ont été précédemment introduit dans la section III. Comme nous l'avons déjà dit, le méta-modèle est la pierre angulaire du cadre technologique grâce aux paramètres d'ingénierie basés sur les modèles [Ken03]. Ce cadre devrait également offrir certaines capacités, comme les capacités structurelles et formelles de vérification afin de garantir un comportement cohérent. Ainsi, le méta-modèle a été enrichi d'un ensemble d'OCL (Object Contraintes Langage [OCL]) règles pour garantir la structure cohérence des instances. De plus, nous prévoyons de proposer un moteur de transformation capable de convertir n'importe quelle instance conforme à ADS^2 à un modèle formel comme les réseaux de Petri pour vérifier les paramètres ainsi que les propriétés de vivacité, de sécurité et de confidentialité. Nous reprenons notre processus dans la Figure 3.5. Pour la mise en œuvre, nous adapterons notre méta-modèle ADS^2 à un standard Méta-Modèle MAS pour tirer parti de l'outil existant, d'où générer directement le code correspondant.

3.7 Intégration du paradigme Skyline dans la prise de décision

Les développements récents des Systèmes Interactifs d'Aide à la Décision (SIAD) ou DSS ont créé de nouvelles situations de prise de décision, qui nécessitent des décisions critiques ce qui oblige les chercheurs à trouver de nouvelles méthodes et algorithmes pour améliorer le comportement des DSS. Les processus actuels souffrent de plusieurs aspects, tels que la difficulté de générer des choix utiles à temps et surtout fiables pour les décideurs.

Cette partie propose une nouvelle approche pour améliorer le processus de décision basée sur les architectures Salsa (Sort and Limit Skyline algorithm) et HDMSM (Hybrid Decision Making Support Model) pour l'extraction de critères afin d'aider les décideurs à choisir les alternatives les plus appropriées dans un délai acceptable. L'avantage de l'approche réside dans l'utilisation du paradigme des requêtes Skyline pour la première fois en tant que processus interne des DSS et sur des tables de décision plutôt que des tables relationnelles classiques. Nous améliorons ainsi la réponse en terme de temps en limitant le processus interne du HDMSM. Enfin, pour valider l'approche proposée, une étude de cas démontre la force de la proposition et montre comment notre approche peut améliorer le temps du processus de décision, en assurer la fiabilité et la qualité du système. Pour garantir la qualité et la performance de la réponse dans les DSS modernes, nous avons intégré le paradigme Skyline dans le processus interne de prise de décision.

3.7.0.1 Motivation de la problématique

La prolifération des données dans les systèmes d'information (SI) a connu une progression exponentielle pour s'adapter aux applications modernes et actuelles. Un des domaines les plus étudiés est celui des DSS, environ 2 400 000 articles sur Google Scholar. Malheureusement, ces systèmes souffrent de la grande quantité et diversité d'informations traitées, par exemple le contexte médical nécessite de nouvelles ressources, méthodes et approches pour assurer des décisions fiables et critiques. Dans les environnements DSS, les résultats doivent respecter le temps et rester persistants en tenant compte de tous les critères définis par les experts [A⁺09].

Dans cette partie, nous proposons B2D (DataBase To Decision tables) une adaptation Skyline pour obtenir une solution optimisée et qui respecte le temps. Actuellement, il n'y a pas d'application ou d'utilisation du modèle Skyline dans le processus DSS, car les architectures DSS existantes utilisent des modèles d'optimisation généralement prédéfinis [A⁺09] pour aider le décideur à résoudre le problème rencontré, le DSS propose plusieurs alternatives basées sur les critères utilisateur (les conditions de la requête) et sur la formulation du problème.

L'opérateur Skyline est utile pour un certain nombre d'applications de bases de données [KLP75]; [KK02]; [BKS01b]; [PE15]; [BCP08a, BCP06]. Skyline peut récupérer des objets présentant la combinaison optimale de préférences utilisateur [PW16], il n'est pas nécessaire de définir une fonction score pour classer les éléments car Skyline se base principalement sur la relation de dominance.

L'analyse Skyline dans les bases de données a été largement étudié en raison de l'énorme avantage qu'apporte cet opérateur. Skyline est très efficace et utile dans ce domaine (environ 195 000 articles scientifiques ont été publiés sur Google Scholar en 2022). Ce type d'analyse ne s'applique pas seulement aux bases de données relationnelles, mais peut aider dans tous les domaines dans lesquels une décision est importante [BKS01b]; [LB13]. Pour cette raison, nous sommes convaincus que cela améliorera considérablement le processus d'un DSS. Skyline n'a jamais été utilisé auparavant dans le processus interne des DSS ni sur les tables de décision (mais uniquement sur des tables de base de données). L'utilisation de cet opérateur a plusieurs avantages, par exemple : l'amélioration des performances (en réduisant la taille de la table considérée au début du processus), la qualité de la décision et la persistance de la réponse (Qui sont des caractéristiques Skyline : la persistance et la fiabilité de la réponse), qui sont très importants pour proposer des alternatives de décision aux décideurs.

Une grande partie des algorithmes Skyline ont des problèmes et ne sont pas tous bien adaptés à ce type de systèmes. Pour cette raison, nous avons d'abord procédé à une analyse critique des algorithmes existants que nous avons présenté dans le chapitre 2 et choisi l'algorithme Salsa [BCP08a, BCP06], qui convient particulièrement à notre problème en termes de performances et fiabilité de réponse.

Notre objectif est de remédier aux faiblesses des DSS, en termes de temps de réponse et qualité de décision. Nous proposons une approche générale et nous nous concentrons sur la troisième étape du processus de prise de décision (choix des meilleures alternatives), comme indiqué dans la figure 3.6, qui est l'une des principales phases prises en charge dans le DSS [FMGG09].

Le processus de prise de décision utilisé est basé sur l'approche HDMSM [WYL15] - un modèle hybride d'aide à la décision permettant d'utiliser le processus d'extraction des critères en cas de conflit. L'approche HDMSM représente et respecte parfaitement le processus standard DSS illustré sur la figure 3.6. Nous améliorons la dernière étape de HDMSM : classement et sélection des alternatifs en utilisant Salsa comme application Skyline.

Comment garantir la qualité et la performance de la réponse ?

- Une approche pour améliorer le processus de décision fondée sur les architectures Salsa et HDMSM, appliquée sur des tables de décision plutôt que des tables relationnelles classiques.
- Une étude de cas qui démontre la faisabilité de la proposition tout en garantissant la fiabilité et la qualité de la décision retournée.

Processus DSS Les étapes principales d'un DSS sont principalement 4 phases, l'intelligence qui consiste à formuler le problème de prise de décision, la conception, qui est une étape qui permet de générer les différentes actions ou alternatives de décision, le choix de la décision, reste une étape critique et importante dans ce processus interne car elle influence directement sur la dernière étape qui est le résultat final. La figure 3.6 présente ces étapes en ordre. Nous nous concentrons sur la troisième étape du processus de prise de décision (choix des meilleures alternatives)

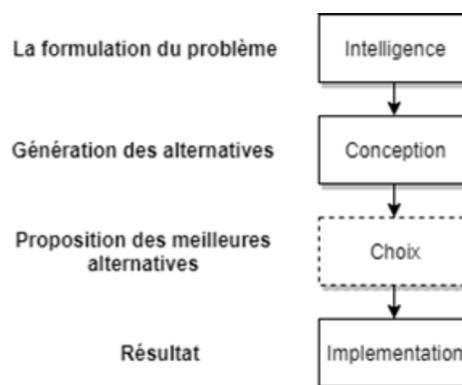


FIGURE 3.6 – Processus général d'un DSS

3.7.0.2 Approche B2D

A partir des points précédents, nous avons proposé une amélioration du processus DSS en intégrant l'algorithme SALSA qui permet d'optimiser le processus de calcul des points Skyline par le biais du point d'arrêt. Voici l'algorithme proposé qui renforce le processus de fonctionnement présenté dans la figure 3.6.

Nous proposons une nouvelle approche nommée B2D. L'approche générale dépend de l'extraction de critères pour appliquer correctement le processus Skyline. On s'inspire de HDMSM (modèle DSS bien valide) pour la définition des critères et Salsa pour l'intégration Skyline. Cette section est composée de trois parties, nous expliquons les comportements HDMSM, Salsa et notre approche B2D.

3.7.0.3 Modèle de décision hybride

Le modèle d'aide à la prise de décision hybride est l'une des approches récentes et bien définies des modèles DSS (en particulier les DSS de groupes), il intègre radicalement 4 approches décisionnelles [WYL15], à savoir Delphi, DEMATEL, ANP et MDS, afin d'aider le décideur à choisir les critères et à établir un classement alternatif face à un problème de décision, elles sont résumées comme suit :

- (1) Identification du critère : Certains critères sont sélectionnés comme critères de décision candidats. Ensuite, les experts établissent un score de consensus pour chaque critère, les

critères de décision les plus importants ayant le poids le plus important étant ensuite choisis parmi les candidats comme critères d'évaluation.

- (2) Calcul de la corrélation des critères : L'approche propose de créer une matrice de corrélation, en fonction de l'importance des critères précédents, DEMATEL évalue la corrélation et calcule le score de prééminence pour chaque critère.
- (3) Évaluation des critères : Un questionnaire d'experts ANP est conçu sur la base d'un cadre d'évaluation. Les experts remplissent le questionnaire ANP, les questionnaires sont ensuite collectés et analysés via la méthode ANP pour en déduire le poids absolu. Après, pour chaque critère de décision, multiplier de manière croisée le poids absolu par le score de consensus (à l'étape 1) et le score de visibilité (à l'étape 2), afin d'obtenir un poids composite important. Enfin, en fonction des poids d'importance composite, HDMSM peut classer les critères de décision.
- (4) Sélection des critères critiques : Pour évaluer la faisabilité de chaque alternative, plusieurs critères de décision sont généralement pris en compte simultanément. Par conséquent, dans cette étape, en fonction des résultats de l'analyse de l'étape 3, un décideur définit un seuil pour le poids d'importance composite.
- (5) Rang alternatif et comparaison : Pour un cas complexe et particulier, une alternative appropriée pourrait être une combinaison de plusieurs alternatives. Par conséquent, le décideur doit comprendre l'effet de chacun des critères d'évaluation. En outre, ils doivent analyser la similarité structurelle entre différentes alternatives. Par conséquent, à l'étape 5, l'analyse MDS est utilisée pour analyser les résultats de l'étape 4 afin d'affecter les positions des alternatives dans une carte perceptuelle d'une deuxième dimension pour présenter la similitude et la non similarité entre les alternatives via la visualisation.

Pour la suite de cette section, nous proposons d'intégrer et d'adapter Salsa en modifiant la dernière phase du processus HDMSM (étape 5) pour obtenir rapidement des alternatives fiables.

3.7.0.4 Algorithme Salsa

L'algorithme Salsa [KT17] est une amélioration de SFS et de LESS présenté dans le tableau 2.2. Il s'efforce d'éviter l'analyse de l'intégralité des données triées, contrairement aux propositions précédentes. C'est le premier algorithme qui exploite les valeurs d'une fonction de notation monotone (limitation) pour trier les données à lire et à comparer. Salsa diffère des autres algorithmes génériques car il limite systématiquement le nombre de points lus et les tests de dominance. La conception de Salsa repose sur deux concepts clés :

premièrement, une étape de tri des données d'entrée et, deuxièmement, choisir de manière appropriée une fonction de tri qui ne privilégie aucun attribut par rapport aux autres (la fonction n'influence pas sur la correction des points Skyline mais seulement sa performance).

Pendant la phase de filtrage, l'algorithme lit et examine les n-tuples une à la fois. Chaque fois qu'un nouveau point est lu, il est comparé à la liste skyline actuelle. Si un point domine le tuple, il est ignoré, sinon il est inséré dans la liste Skyline (en tant que point Skyline final) et l'algorithme vérifie son déclencheur de terminaison (P_{stop}). Si le seuil actuel P_{stop} est inférieur ou égal à la valeur f_{min} du point, l'algorithme se termine et renvoie les points Skyline complets. Cette condition de terminaison garantit qu'aucun point de données examiné ultérieurement ne doit faire partie de la liste Skyline. L'algorithme évite ainsi d'analyser l'intégralité des données.

Il existe de nombreuses fonctions limitantes dans la littérature, la fonction optimale permettant de limiter une relation d'entrée plus que d'autres est la fonction de coordonnée minimale (comparée à Sum et à Val présentée dans le travail de [BCP08a, BCP06]). Noté MinC : la fonction

trie d'abord les points en tenant compte de la valeur de coordonnées minimale du point actuel et simultanément, la fonction Somme est calculée en cas d'égalité.

Motivation : L'algorithme est efficace, lorsqu'il traite une grande quantité de données, c'est le plus adaptatif par rapport aux autres algorithmes en raison de l'importance temporelle des systèmes DSS spécialement pour les décisions critiques. Salsa est progressive, ce qui signifie que nous n'avons pas besoin de refaire le traitement pour obtenir des points Skyline exacts (les résultats sont progressifs), c'est le cas des DSS, nous avons besoin de réponses progressives pour éviter de bloquer le processus de décision. Une autre raison de choisir Salsa est l'absence d'indexation. Par conséquent, il n'a pas besoin d'un traitement préalable pour fonctionner ou d'utiliser une structure de données particulière.

3.7.0.5 Processus B2D

Le Skyline a toujours été utilisé pour traiter des requêtes générales sur des bases de données relationnelles, nous nous appuyons sur le mécanisme Skyline pour l'appliquer aux tables de décision et comme ça obtenir les meilleures décisions à l'aide des critères choisis, extraites après utilisation de HDMSM. Les tables de décision sont similaires aux tables de base de données relationnelles. Au lieu d'avoir des valeurs, nous avons des conditions et chaque tuple représente une action qui peut être exécutée si toutes les conditions sont satisfaites [Van10]; [VW95], la différence reste dans les types de valeur comme les conditions composées. Dans les DSS, la décision dépend de plusieurs critères contradictoires. C'est la raison pour laquelle nous avons choisi Skyline afin de sélectionner les actions (décisions) appropriées pour les décideurs en tenant compte de manière équitable tous les critères définis par les experts.

B2D comprend deux parties, HDMSM étant utilisé pour formuler le problème avec un ensemble de critères, B2D a besoin de deux acteurs (le décideur et les experts). La deuxième étape consiste à générer des alternatives basées sur le problème formulé et extrait de la première étape et sur les tables de décision existantes.

Les phases de l'approche sont résumées (en fonction des différents acteurs) comme suit :

— **Le décideur**

1. Formule le problème de la décision (détermine le but de la prise de décision)

— **Les experts**

2. Récupèrent les critères (avec le poids des experts)
3. Listent les critères d'évaluation pertinents (questionnaires de groupe et consensus)

— **Génération d'alternatives (en utilisant Salsa)**

Les étapes 1, 2 et 3 sont basées sur le modèle hybride d'aide à la décision. Après l'étape 3, nous aurons comme résultat : un objectif de décision et des alternatives générées avec leurs critères. À l'étape 4, nous appliquons Skyline (algorithme adaptatif de Salsa) aux alternatives générées (où les points deviennent des choix de décision). Un calcul détaillé de B2D est résumé dans la figure 3.7 et le processus suivant illustre les étapes essentielles de B2D utilisées pour extraire les décisions.

B2D DSS basé Skyline
1: Entrée : Extraction de critères des experts & Formulation du problème → Requête
2: Pré-traitement des données → Nettoyage des données et extraction des connaissances
3: Normalisation → Formule :
$f(p.d_i) = (p.d_i - min_i)/(max_i - min_i)$
4: Trier → MinC et Sum (si nécessaire contrairement à l’algorithme Salsa)
5: Application du Skyline → Processus de domination Salsa
6: Sortie : Des points Skyline → des décisions fiables

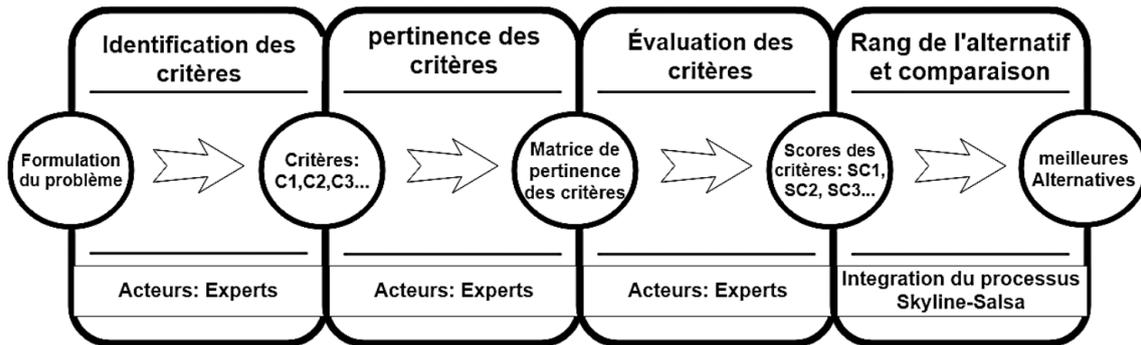


FIGURE 3.7 – Différentes étapes de l’approche proposée

3.7.0.6 Scénario et implémentation de l’algorithme B2D (étude de cas)

Pour la phase expérimentale, nous avons préféré appliquer notre approche à un scénario concret dans un contexte médical afin d’illustrer le fonctionnement et utiliser cette approche. Imaginons le scénario suivant illustré à la figure. 3.9 : La proposition d’un traitement adéquat à un patient (mais avec un certain degré d’incertitude), nous avons les analyses du patient après avoir consulté le médecin et ses antécédents (dossiers médicaux). Sur la base de ces deux données d’entrée, le médecin propose un ensemble de traitements au patient (mais avec un certain degré d’incertitude à diminuer). La tâche consiste à trouver les traitements les plus appropriés en fonction du patient. Le DSS contient une base de connaissances interne ou table de décision avec symptômes et un ensemble de traitements associés à ces symptômes (conditions-action). Le processus consiste à appliquer Skyline sur les tables de décision après avoir déterminé les critères de décision, le système propose ou corrige les décisions du médecin.

Nous avons utilisé "Generate Data" sur un serveur local pour générer des données réelles. B2D a été développée sous Eclipse IDE, Version : Oxygen.3 Release (4.7.3) IDE, en utilisant le langage Java.v9, Xampp (MySQL) et des fichiers (CSV). L’algorithme a été testé sur une machine Dell Inc, modèle système : Precision T1650, et exécuté dans un environnement Windows 10 Éducation 64 bits (10.0, Build 16299), utilisant un processeur Intel(R) Core(TM) i7-3770 3.4GHz @ 3.40 GHz (8 CPU) avec 8Go de RAM, en mode séquentiel (1 thread).

Nous avons effectué deux tests principaux : Modifier la taille et les choix de décision (propositions du médecin). Chaque test a été lancé $1 * 10^8$ pour généraliser les résultats, nous avons calculé l’écart type qui était très petit pour les 2 tests, puis la valeur moyenne.

Test 1 : comme indiqué dans le tableau. 2, nous avons testé l’évolution du temps d’exécution qui dépend de la taille de la table de décision. Plus précisément, lorsque nous avons :

— 1 patient avec son analyse et 10^2 tuples (9Ko pour le format CSV) de la base de connaissances (table de décision - traitement ou actions si les conditions sont remplies), l'objectif est de minimiser la satisfaction des conditions pour obtenir les actions les plus adaptées à un patient spécifique

— 1 patient avec un tuples de 10^3 (82Ko)

— Les mêmes données pour 10^4 tuples (815Ko)

— Pour le dernier, nous augmentons à 10^5 (8Mo)

Test 2 : comme indiqué dans le tableau. 2, les mêmes variations avec une différence dans les choix de décision (propositions), les patients et la taille des analyses :

— 5 patients, 5 analyses, 10 propositions par patient, avec un choix de 10 2 dans la table de décision

— 10 patients, 10 analyses, 15 propositions par patient, avec une table de décision de 10^3

— 15 patients, 15 analyses, 20 propositions par patient, avec une table de décision de 10^4

— 20 patients, 20 analyses, 25 propositions par patient, avec une table de décision de 10^5

Ces tests ont été effectués pour illustrer la faisabilité de la proposition. Le pré-traitement est une étape nécessaire dans notre approche pour obtenir un critère de sélection valide (dans cet exemple, nous minimisons par exemple la différence entre les symptômes du patient et la valeur de la connaissance du symptôme. plus la différence est petite, plus le traitement est adapté au patient).

Discussion : Les tests portent sur deux symptômes (2 dimensions), les premiers résultats montrent que cette approche est très bénéfique en termes de performances, de fiabilité de réponse et qualité de décision, l'utilisation de l'opérateur Skyline garantit que la démarche retourne toujours les meilleurs choix aux décideurs (mécanisme de dominance pareto).

En observant Test 1 et 2, le temps d'exécution dépend de la taille des tables de décision et de la formulation de la requête (formulation du problème), les résultats initiaux sont plutôt positifs (nous avons volontairement augmenté le nombre d'entrées pour valider B2D, test 1 : le temps d'exécution ne dépasse pas 5 millisecondes, test 2 : la durée d'exécution est inférieure à 206 ms).

Le résultat de l'implémentation de cet algorithme est présenté dans la figure 3.9 où nous avons constaté que notre approche améliore le processus interne en réduisant le temps d'exécution même si on augmente la taille des tables de décision.

Une mise en œuvre de cette proposition étant établie, on distingue deux cas principaux :

— Nous résolvons le problème lorsque les données sont petites, ce qui est parfois le cas d'un système décisionnel (nous n'avons pas nécessairement une masse de données).

— Nous augmentons ensuite progressivement la taille de la table de décision et les entrées (conditions).

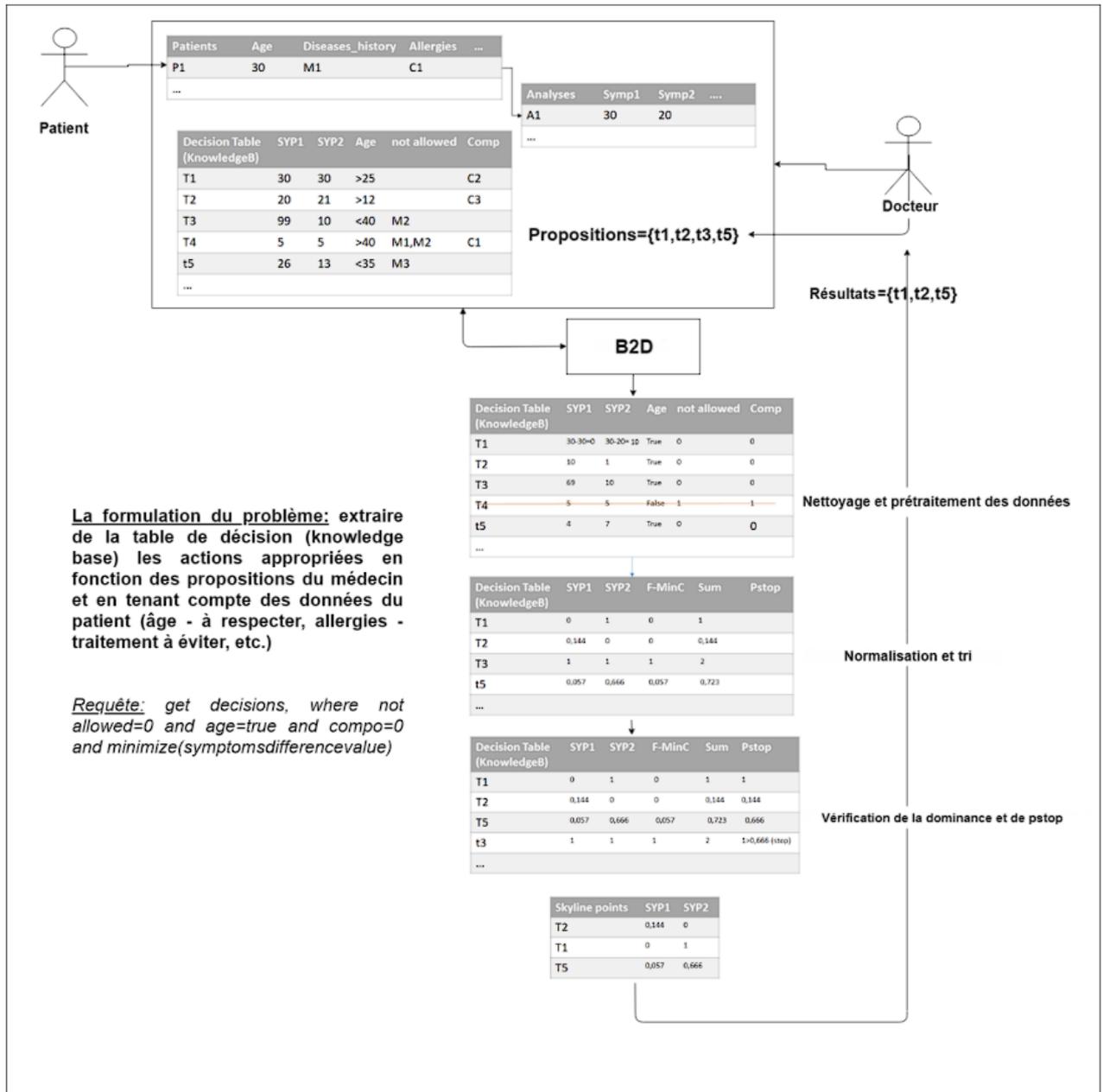


FIGURE 3.8 – Scénario d'étude de cas

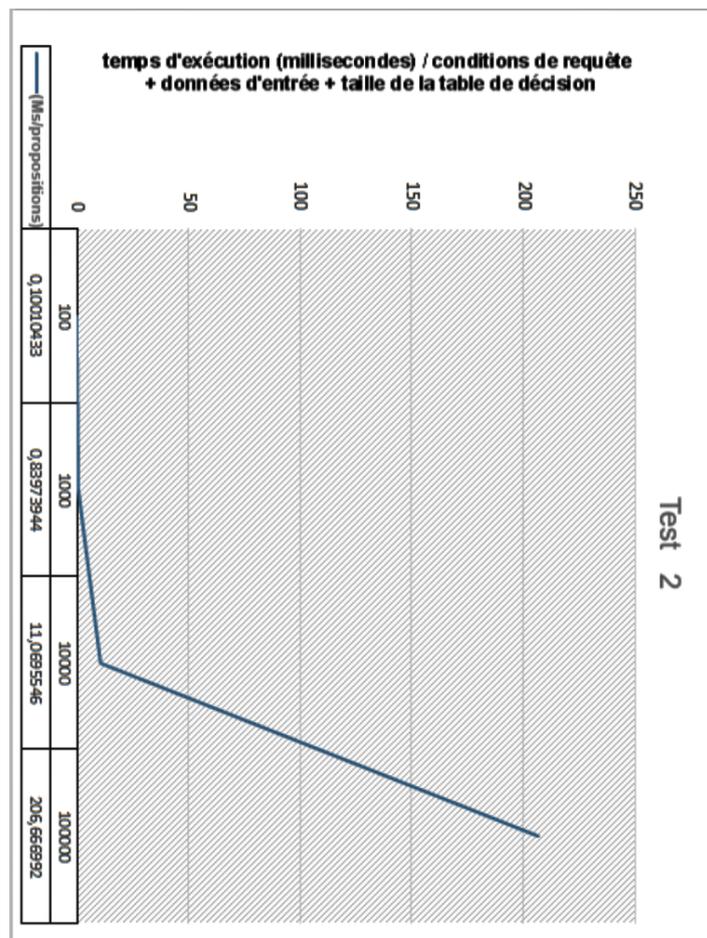
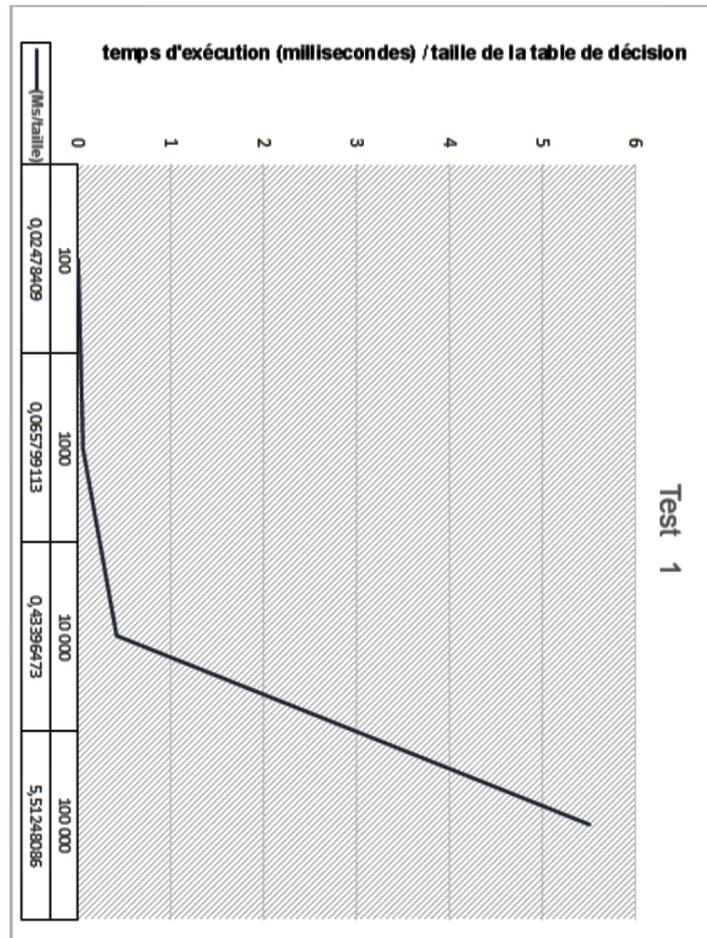


FIGURE 3.9 – Temps d'exécution de B2D

3.8 Conclusion et Bilan

Dans ce chapitre, nous avons abordé la modélisation d'un DDSS à l'aide du paradigme MAS en présentant une étude comparative faisant référence à plusieurs critères. De ce fait, notre proposition était de répondre aux différents critères trouvés dans les travaux précédents. Ensuite, nous avons défini une nouvelle approche MAS pour DDSS appelée *ADS²*, en intégrant (i) les nouveaux agents miroirs pour augmenter la tolérance aux pannes et (ii) l'agent observateur pour gérer les contraintes d'environnement. Nous avons également proposé d'affiner le processus DSS en utilisant (iii) des agents BDI.

Pour répondre à l'une des faiblesses du processus DDSS interne, nous avons proposé l'approche B2D qui exploite le paradigme du Skyline. Ensuite, nous avons mis en œuvre et vérifié notre proposition dans différentes situations (taille/diversité/cas), les premiers résultats sont très prometteurs.

Chapitre 4

Approche distribuée et raffinée du modèle à base de SMA

Sommaire

4.1	Introduction	65
4.2	Définition de la problématique	66
4.3	Relaxation du Skyline de groupe	66
4.3.1	Dominance de groupe floue	66
4.3.2	Méthode RG-SKY	68
4.3.3	Propriétés de base	69
4.3.4	Calcul du RG-SKY	70
4.4	Étude expérimentale	74
4.4.1	Environnement expérimental	74
4.4.2	Tests expérimentaux	74
4.4.3	Résultats expérimentaux	75
4.5	Conclusion et Bilan	81

4.1 Introduction

Le modèle DDSS fondé sur l'approche SMA présenté dans le chapitre 3, sera augmenté et doté par un processus interne basé sur le Skyline de groupe optimisé qui permet de donner une réponse raffinée et robuste.

Depuis l'introduction des requêtes skyline dans le domaine des bases de données par Borzsonyi et al.[BKS01a] en 2001, un grand effort de recherche a été consacré au développement d'algorithmes efficaces pour le calcul du skyline de groupe. Le calcul du skyline de groupe pour une prise de décision peut mener à deux scénarios : soit (i) un grand nombre de groupe skyline qui est moins informatif pour les utilisateurs finaux ou (ii) un petit nombre de groupe skyline qui peut être insuffisants pour les besoins de l'utilisateur. Afin de résoudre ces deux problèmes, plusieurs travaux ont été effectués soit pour raffiner le skyline de groupe en diminuant sa taille par contre le deuxième scénario n'a pas été traité par les scientifiques pour relaxer le groupe skyline en augmentant sa taille.

Dans le scénario (ii), relaxer le Skyline de groupe permet d'augmenter le nombre de résultats de groupes Skyline et ainsi mieux satisfaire les besoins des décideurs. La solution est d'utiliser une variante relaxée du Skyline de groupe, cette variante est dirigée par les préférences utilisateurs et permet de faire un véritable contrôle des résultats retournés. Considérons le problème suivant consistant à trouver les groupes Skyline (où l'indique le nombre d'éléments dans chaque groupe à partir d'un ensemble de données) : un décideur (le formateur) veut obtenir les meilleurs 5 groupes de joueurs. 3 en maximisant les points et le nombre de blocs. Le système renvoie 2 groupes de 3 joueurs de volley chacun. Malheureusement, cette réponse ne satisfait pas le décideur, il/elle a besoin de 5 équipes des meilleurs joueurs mais la définition traditionnelle de Skyline de groupe ne peut renvoyer que 2 groupes. C'est la question principale abordée dans ce chapitre. La solution préconisée vise à élargir la taille de Skyline du groupe en appliquant un processus de relaxation approprié, en utilisant le principe de la logique floue.

Le reste de ce chapitre est organisé comme suit. Dans la Section 1, nous présentons la problématique en détaille. La Section 2 et 3 sont consacrées à la présentation de l'approche RG-SKY et les algorithmes du contrôle de taille du skyline, c-à-d les algorithmes de la relaxation. La section 4, présente une étude expérimentale détaillé de notre approche. La Section 5 conclut ce chapitre.

4.2 Définition de la problématique

Soit Q une requête Skyline utilisateur et D l'ensemble de données cible. Supposons que l'utilisateur souhaite trouver K groupes d'éléments à des fins de décision. Soit $G\text{-SKY}(D)$ le skyline de groupe calculé et $|G\text{-SKY}(D)|$ être sa taille.

On peut facilement observer que si $|G\text{-SKY}(D)| < K$, l'utilisateur n'est alors pas en mesure de prendre la décision souhaitée en raison de l'insuffisance des groupes skyline retournés. Le problème d'intérêt est alors de savoir comment agrandir la taille de $G\text{-SKY}(D)$ afin d'obtenir une variante relaxée, appelée $RG\text{-SKY}(D)$, avec plus de groupes (ie, $G\text{-SKY}(D) \subseteq RG\text{-SKY}(D)$). Nous appelons ce problème le problème de relaxation de skyline de groupe.

4.3 Relaxation du Skyline de groupe

Nous discutons ici notre approche floue pour relaxer le Skyline de groupe. L'idée est d'étendre la domination du groupe (donnée dans la définition 2.3.1 de $G\text{-Skyline}$ dans le chapitre 2) en le rendant plus exigeant. Le groupe Skyline détendu obtenu, $RG\text{-SKY}$, n'est plus un ensemble plat mais un ensemble discriminé où chacun de ses éléments est associé à un degré.

Dans toute la suite du chapitre, nous utilisons les symboles et notations données dans le tableau 2.1 du chapitre 2.

L'idée principale consiste à calculer dans quelle mesure un groupe, écarté par la relation de dominance $G\text{-Skyline}$, peut appartenir au Skyline de groupe relaxé. A cette fin, et comme il sera illustré plus loin, nous associons à chaque attribut de Skyline $A_i (i \in \{1, \dots, d\})$ une paire de paramètres $(\gamma_{i1}, \gamma_{i2})$ où γ_{i1} et γ_{i2} désignent respectivement les bornes de la zone de relaxation autorisée à l'attribut A_i . Un vecteur de paires de paramètres, noté γ , est alors défini par la formule suivante :

$$\gamma = ((\gamma_{11}, \gamma_{12}), \dots, (\gamma_{d1}, \gamma_{d2})).$$

Il est à noter que γ , appelé vecteur de paramètre de relaxation, est un vecteur défini par l'utilisateur¹, ce qui signifie que l'approche est centrée utilisateur. Il définit l'ensemble de valeurs, pour chaque attribut que l'utilisateur peut tolérer même s'il est exclu lors de l'application de la dominance \prec_{gs} .

4.3.1 Dominance de groupe floue

$RG\text{-SKY}$, le skyline de groupe relaxé de $G\text{-SKY}$, s'appuie sur une relation de dominance particulière qui permet d'élargir le skyline de groupe avec les groupes les plus intéressants parmi ceux exclus lors du calcul de $G\text{-SKY}$ utilisant la définition 2.3.1 de $G\text{-Skyline}$ présentée dans le chapitre 2. Cette relation de dominance utilise la relation floue "Much Preferred ($MP_{\mathbb{G}}$)" pour comparer deux groupes g et g' .

Ainsi, g est un élément de $RG\text{-SKY}$ s'il n'y a pas de groupe $g' \in G$ tel que g' est fortement préféré à g (noté $MP_{\mathbb{G}}(g', g)$) sur tous les attributs du skyline de groupe. Formellement, on écrit :

$$g \in RG - SKY \Leftrightarrow \nexists g' \in \mathbb{G}, MP_{\mathbb{G}}(g', g) \quad (4.1)$$

1. L'utilisateur prédéfinit les valeurs ou le degré de tolérance de chaque dimension sous la forme d'un vecteur de relaxation.

Notons que g' est fortement préféré à g dans le sens de $MP_{\mathbb{G}}$ si et seulement si g' est fortement préféré à g c'est à dire sur toutes dimension i in $\{1, \dots, d\}$. Formellement, on écrit :

$$MP_{\mathbb{G}}(g', g) \Leftrightarrow \forall i \in \{1, \dots, d\}, MP_{\mathbb{G}_i}(g'(i), g(i)) \quad (4.2)$$

où $MP_{\mathbb{G}_i}$ est définie sur le domaine D_i de l'attribut A_i , $g(i) = F(Q_1[i], \dots, Q_l[i])$ (resp. $g'(i) = F(Q'_1[i], \dots, Q'_l[i])$) et F est une fonction d'agrégation.

Rappelons que $(g'(i), g(i)) \in MP_{\mathbb{G}_i}$ signifie que $\mu_{MP_{\mathbb{G}_i}}(g'(i), g(i)) > 0$ (ou simplement $MP_{\mathbb{G}_i}(g'(i), g(i)) > 0$). De la même manière, $(g', g) \in MP_{\mathbb{G}}$ signifie également $\mu_{MP_{\mathbb{G}}}(g', g) > 0$ (ou simplement $MP_{\mathbb{G}}(g', g) > 0$).

Notons que $MP_{\mathbb{G}_i}(g'(i), g(i))$ exprime dans quelle mesure la valeur $g'(i)$ est fortement préférée à la valeur de $g(i)$. Comme $MP_{\mathbb{G}_i}$ est de nature graduelle, chaque élément g de RG-SKY est associé à un degré ($\in [0, 1]$) exprimant la mesure dans laquelle g appartient à RG-SKY. En terme d'ensembles flous, on peut écrire :

$$\mu_{RG-SKY}(g) = 1 - \max_{g' \in \mathbb{G} - \{g\}} \min_i \mu_{MP_{\mathbb{G}_i}}(g'(i), g(i)) = \min_{g' \in \mathbb{G} - \{g\}} \max_i (1 - \mu_{MP_{\mathbb{G}_i}}(g'(i), g(i))) \quad (4.3)$$

La sémantique de la relation floue $MP_{\mathbb{G}_i}$ peut être exprimée par la formule 4.4 (voir aussi figure 4.1).

$$\mu_{MP_{\mathbb{G}_i}^{(\gamma_{i1}, \gamma_{i2})}}(g'(i), g(i)) = \begin{cases} 0 & \text{si } g'(i) - g(i) \leq \gamma_{i1} \\ 1 & \text{si } g'(i) - g(i) \geq \gamma_{i2} \\ \frac{(g'(i) - g(i)) - \gamma_{i1}}{\gamma_{i2} - \gamma_{i1}} & \text{Autrement} \end{cases} \quad (4.4)$$

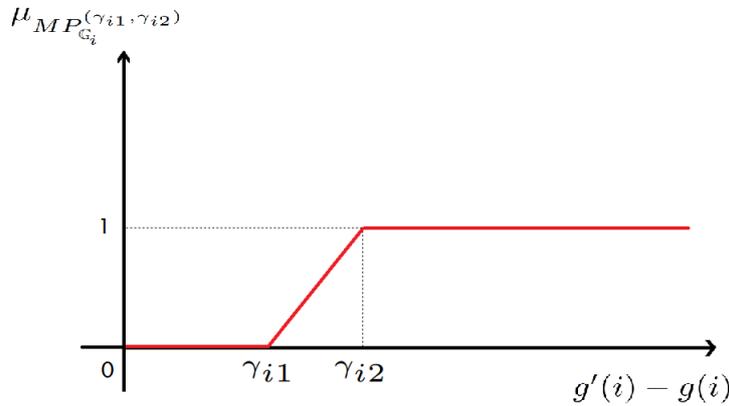


FIGURE 4.1 – La fonction d'appartenance $MP_{\mathbb{G}_i}^{(\gamma_{i1}, \gamma_{i2})}$

Par exemple, si $g'(i) - g(i) \geq \gamma_{i2}$ alors $g'(i)$ est complètement préféré à $g(i)$. On peut aussi voir que si $g'(i) - g(i) > \gamma_{i1}$, $g'(i)$ est non seulement préféré (*preferred*) mais bien préféré (*much preferred*) que $g(i)$ dans une certaine mesure. L'ensemble flou associé à $MP_{\mathbb{G}_i}$ s'écrit $(\gamma_{i1}, \gamma_{i2}, \infty, \infty)$, et noté $MP_{\mathbb{G}_i}^{(\gamma_{i1}, \gamma_{i2})}$. Il est facile de vérifier que $MP_{\mathbb{G}_i}^{(0,0)}$ correspond à la relation de préférence booléenne s'exprimant au moyen de la relation régulière "supérieur à" ("*greater than*").

Soit maintenant $RG_SKY^{(\gamma)}$ le Skyline de groupe relaxé calculé sur la base du vecteur de relaxation $\gamma = ((\gamma_{11}, \gamma_{12}), \dots, (\gamma_{d1}, \gamma_{d2}))$ dans le cas de d attribut Skyline.

On peut facilement vérifier que le groupe skyline $G - SKY = RG_SKY^{(\mathbf{0})}$ avec $\mathbf{0} = ((0, 0), \dots, (0, 0))$.

On peut aussi vérifier que la propriété de monotonie suivante est vérifiée.

Proposition 1 Soient γ and γ' deux vecteurs de paramètres de relaxation. Alors, la propriété suivante est vraie :

$$\gamma' \leq \gamma \Rightarrow RG - SKY(\gamma') \subseteq RG - SKY(\gamma)$$

PROOF , Soient $\gamma = ((\gamma_{11}, \gamma_{12}), \dots, (\gamma_{d1}, \gamma_{d2}))$ et $\gamma' = ((\gamma'_{11}, \gamma'_{12}), \dots, (\gamma'_{d1}, \gamma'_{d2}))$ deux vecteurs de paramètres de relaxation. $\gamma' \leq \gamma \Rightarrow \forall i \in \{1, \dots, d\} \gamma'_{i1} \leq \gamma_{i1}$ et $\gamma'_{i2} \leq \gamma_{i2}$. Cela implique que $\forall i \in \{1, \dots, d\} MP_{\mathbb{G}_i}^{(\gamma_{i1}, \gamma_{i2})} \subseteq MP_{\mathbb{G}_i}^{(\gamma'_{i1}, \gamma'_{i2})}$. Based on (4.3), on peut en déduire que $RG - SKY(\gamma') \subseteq RG - SKY(\gamma)$ est vérifiée.

Lemma 1 Soient $\gamma = ((0, \gamma_{12}), \dots, (0, \gamma_{d2}))$, $\gamma' = ((\gamma'_{11}, \gamma'_{12}), \dots, (\gamma'_{d1}, \gamma'_{d2}))$ and $\forall i \in \{1, \dots, d\} \gamma_{i2} < \gamma'_{i2}$, le résultat suivant est également valable :

$$RG - SKY(0) \subseteq RG - SKY(\gamma) \subseteq RG - SKY(\gamma')$$

Exemple 3. Pour illustrer l'intérêt de l'ensemble RG-SKY, considérons un exemple simple de 4 hôtels avec des valeurs proches comme illustré dans le tableau 4.1. Supposons que l'utilisateur veuille récupérer les K=3 meilleurs groupes d'hôtels en minimisant le deux dimensions le prix et la distance, pour obtenir les hôtels les plus proches et les moins chers. Pour cela, nous procédons comme suit :

Hôtels	Prix (Euro)	Distance (Km)
h_1	50	10
h_2	100	6
h_3	150	5
h_4	40	11

TABLEAU 4.1 – Exemple d'utilisation de Skyline pour trouver les meilleurs hôtels pour une conférence donnée

- **Étape 1 : Calcul du Skyline** Générez les points Skyline : il est facile de vérifier $SKY = \{h_1, h_2, h_3, h_4\}$
- **Étape 2 : Génération de G** Générer les groupes de taille 3 à l'aide du coefficient binomial² et appliquer la fonction d'agrégation (F_{min}) :
 - $g_1 = \{h_2, h_3, h_4\}, F_{min}(g_1) = \langle \min(100, 150, 40), \min(6, 5, 11) \rangle = \langle 40, 5 \rangle$
 - $g_2 = \{h_1, h_3, h_4\}, F_{min}(g_2) = \langle \min(50, 150, 40), \min(10, 5, 11) \rangle = \langle 40, 5 \rangle$
 - $g_3 = \{h_1, h_2, h_4\}, F_{min}(g_3) = \langle \min(50, 100, 40), \min(10, 6, 11) \rangle = \langle 40, 6 \rangle$
 - $g_4 = \{h_1, h_2, h_3\}, F_{min}(g_4) = \langle \min(50, 100, 150), \min(10, 6, 5) \rangle = \langle 50, 5 \rangle$
- **Étape 3 : Calcul du Skyline de groupe** En appliquant la définition 2.3.1 du chapitre 2, on peut vérifier que skyline du groupe est $G-SKY = \{g_1, g_2\}$.

Malheureusement, l'utilisateur reçoit 2 groupes skyline même si tous les tuples sont des points skyline alors qu'il a besoin de 3 groupes pour prendre une décision. Afin de répondre au besoin initial de l'utilisateur, nous faisons appel alors à l'approche de relaxation de G-SKY.

4.3.2 Méthode RG-SKY

Supposons d'abord que le vecteur de relaxation est $\gamma = ((0.5, 1), (0.5, 1))$, c'est à dire, (0.5, 1) pour les deux attribues Skyline "Prix" et "Distance". D'après l'équation 4.4, on peut vérifier que la relation floue $MP_{\mathbb{G}_{Prix}}$ peut s'écrire :

2. Le coefficient binomial est noté $C(n, k)$ et se lit "choisir k parmi n " et est défini par la formule $C(n, k) = n! / (k!(n-k)!)$ avec $n!$ représente la factorielle de n .

$$\mu_{MP_{\mathbb{G}Prix}^{(0.5,1)}}(v, u) = \begin{cases} 0 & \text{si } v - u \leq 0.5 \\ 1 & \text{si } v - u \geq 1 \\ \frac{v-u-0.5}{1-0.5} & \text{Autrement} \end{cases} \quad (4.5)$$

La relation floue $MP_{\mathbb{G}Distance}$ peut également s'écrire d'une manière similaire.

$$\mu_{MP_{\mathbb{G}Distance}^{(0.5,1)}}(v, u) = \begin{cases} 0 & \text{si } v - u \leq 0.5 \\ 1 & \text{si } v - u \geq 1 \\ \frac{v-u-0.5}{1-0.5} & \text{Autrement} \end{cases} \quad (4.6)$$

Calculons maintenant l'ensemble flou RG-SKY (où $i = 1$ et $i = 2$ désignent respectivement les attributs *prix* et *Distance*) :

$$\begin{aligned} \mu_{RG-SKY}(g_3) &= 1 - \max_{g' \in \{g_1, g_2, g_4\}} \min_{i \in \{1, 2\}} \mu_{MP_{\mathbb{G}_i}}(g'(i), g_3(i)) \\ \mu_{RG-SKY}(g_3) &= 1 - \max[\min(\mu_{MP_{\mathbb{G}_1}}(g_1(1), g_3(1)), \mu_{MP_{\mathbb{G}_2}}(g_1(2), g_3(2))), \min(\mu_{MP_{\mathbb{G}_1}}(g_2(1), \\ &g_3(1)), \mu_{MP_{\mathbb{G}_2}}(g_2(2), g_3(2))), \min(\mu_{MP_{\mathbb{G}_1}}(g_4(1), g_3(1)), \mu_{MP_{\mathbb{G}_2}}(g_4(2), g_3(2)))] \\ \mu_{RG-SKY}(g_3) &= 1 - \max[\min(0, 0), \min(0, 0), \min(1, 0)] = 1 - 0 = 1 \end{aligned}$$

De façon similaire, on obtient $\mu_{RG-SKY}(g_4) = 1$. Alors,

L'ensemble RG-Sky S'écrit : $RG-SKY = \{1/g_1, 1/g_2, 1/g_3, 1/g_4\}$.

On observe que RG-SKY contient les éléments G-SKY (i.e., g_1 et g_2 avec un degré égal à 1) et quelques nouveaux groupes qui n'étaient pas dans G-SKY avec un degré égal à 1 (i.e., g_3 et g_4). RG-SKY est plus grand que G-SKY et peut satisfaire la requête initiale de l'utilisateur en renvoyant par exemple les groupes g_1 , g_2 et g_3 .

Maintenant, si RG-SKY contient plus de K groupes, les K meilleurs groupes sont renvoyés. En cas d'égalité, l'utilisateur peut établir un classement sur la base des préférences des attributs Skyline (dans notre cas, les attributs *Prix* et *Distance*). Comme pour le cas où RG-SKY contient moins de groupes K, on peut réviser le vecteur de relaxation et ré-exécuter la méthode RG-SKY.

4.3.3 Propriétés de base

Nous établissons ici un ensemble de propositions souhaitables qui présentent un intérêt sur le plan calculatoire. Certains d'entre elles sont les contreparties floues des propriétés du Skyline de groupe G-SKY, introduites dans [IP12b].

Soient $\gamma = ((\gamma_{11}, \gamma_{12}), \dots, (\gamma_{d1}, \gamma_{d2}))$ et $\gamma' = ((\gamma'_{11}, \gamma'_{12}), \dots, (\gamma'_{d1}, \gamma'_{d2}))$ $MP_{\mathbb{G}}^{\gamma}(g, g') > 0$ ce qui signifie que $(g, g') \in MP_{\mathbb{G}}^{\gamma}$, on peut définir les propositions suivantes :

Proposition 2 (*Min-Asymétrie*) Soient g et g' deux groupes de \mathbb{G} ,

$$\text{Si } (g, g') \in MP_{\mathbb{G}}^{\gamma} \text{ alors } (g', g) \notin MP_{\mathbb{G}}^{\gamma}.$$

Preuve. La proposition 2 peut aussi s'écrire sous la forme : Si $MP_{\mathbb{G}}^{\gamma}(g, g') > 0$ alors $MP_{\mathbb{G}}^{\gamma}(g', g) = 0$. du fait de la propriété d'asymétrie des préférences floues [PR98], on peut écrire : $\min(MP_{\mathbb{G}_i}^{(\gamma_{i1}, \gamma_{i2})}(g(i), g'(i)), MP_{\mathbb{G}_i}^{(\gamma'_{i1}, \gamma'_{i2})}(g'(i), g(i))) = 0, \forall i \in \{1, \dots, d\}$. A savoir, si, $MP_{\mathbb{G}_i}^{(\gamma_{i1}, \gamma_{i2})}(g(i), g'(i)) > 0$ alors $MP_{\mathbb{G}_i}^{(\gamma'_{i1}, \gamma'_{i2})}(g'(i), g(i)) = 0$.

Proposition 3 (*Min-Transitivity*) Soient g , g' et g'' trois groupes de \mathbb{G} ,

$$\text{Si } (g, g') \in MP_{\mathbb{G}}^{\gamma} \text{ et } (g', g'') \in MP_{\mathbb{G}}^{\gamma'} \text{ alors } (g, g'') \in MP_{\mathbb{G}}^{\gamma+\gamma'}.$$

Preuve. La proposition 3 peut s'écrire sous la forme : Si $MP_{\mathbb{G}}^{\gamma}(g, g') > 0$ et $MP_{\mathbb{G}}^{\gamma'}(g', g'') > 0$ alors $MP_{\mathbb{G}}^{\gamma+\gamma'}(g, g'') > 0$. à présent, [PR98] due à la propriété de transitivité des préférences floues et à l'addition floue, [DKMP00], on peut écrire :

$$\min(MP_{\mathbb{G}_i}^{(\gamma_{i1}, \gamma_{i2})}(g(i), g'(i)), MP_{\mathbb{G}_i}^{(\gamma'_{i1}, \gamma'_{i2})}(g'(i), g''(i))) \leq MP_{\mathbb{G}_i}^{(\gamma_{i1}+\gamma'_{i1}, \gamma_{i2}+\gamma'_{i2})}(g(i), g''(i)), \forall i \in \{1, \dots, d\}$$

Proposition 4 Si $g \subset SKY(\mathbb{D})$ alors $g \in RG-SKY(\mathbb{D})$ ne tient pas toujours.

Preuve. Pour montrer que cette proposition n'est pas toujours vraie, il suffit d'exposer un contre-exemple. Considérons un jeu de données bidimensionnel (points, rebonds) de 4 joueurs : $p_1 = (0,4)$, $p_2 = (1,2)$, $p_3 = (2,1)$ et $p_4 = (4, 0)$. Il est facile de voir que $SKY = \{p_1, p_2, p_3, p_4\}$. Let $g = \{p_1, p_4\} \subset SKY$. On peut vérifier que $g \notin RG-SKY$ pour $\gamma = ((0, 1), (0, 1))$.

Proposition 5 (L'inverse de la proposition 4) Si $g \in RG-SKY(\mathbb{D})$ alors $g \subset SKY(\mathbb{D})$ ne tient pas toujours.

Preuve. Trouvons aussi un contre-exemple. Considérons un jeu de données bidimensionnel (points, rebonds) de 3 joueurs : $p_1 = (0, 2)$, $p_2 = (1, 0)$ et $p_3 = (2,1)$, nous avons $SKY = \{p_3\}$ et $G-SKY = \{\{p_1, p_3\}\}$ (c'est-à-dire contient 1 groupe de 2 points). Pour $\gamma = ((0, 1), (0, 1))$, on peut vérifier que $\{p_2, p_3\} \in RG - SKY$ tandis que $\{p_2, p_3\} \notin SKY$.

4.3.4 Calcul du RG-SKY

Le calcul du RG-SKY devrait faire partie d'un système d'aide à la décision (DSS) [NOSH18]. Ici, nous fournissons (i) un diagramme qui donne un aperçu de la façon dont l'approche RG-SKY est mise en revue et (ii) les deux algorithmes proposés pour le calcul de l'ensemble RG-SKY.

Diagramme RG-SKY

La figure 4.2 donne un aperçu de l'approche RG-SKY et illustre la chronologie de ses étapes de manière globale. Deux cas peuvent être distingués :

En cas 1 : Satisfaction de la demande de l'utilisateur - Détente inutile

Dans ce cas, le décideur envoie une demande avec un ensemble de conditions contradictoires au système DSS (qui intègre le processus de calcul du G-SKY), le traditionnel. Le calcul G-SKY renvoie une réponse satisfaisante et le processus RG-SKY n'est alors pas déclenché.

En cas 2 : Insatisfaction de la demande de l'utilisateur - Relaxation requise.

Le calcul traditionnel de G-SKY renvoie une réponse insatisfaisante. Avant de commencer le processus de relaxation, le système obtient d'abord le vecteur de paramètres de relaxation de l'utilisateur. Le système exécute également des techniques de prétraitement et d'élagage pour accélérer le calcul du RG-SKY. Par exemple, les stratégies d'optimisation suivantes sont mises en œuvre :

- **Tri des groupes par leur nombre de points Skyline :** cela permet de créer une hiérarchie des groupes qui aide à parcourir intelligemment l'espace de recherche.
- **Techniques de taille :** elles s'appuient sur les propositions et propriétés préalablement établies.

RG-SKY représente un ensemble flou dans le sens où chaque groupe est associé à un degré (exprimant dans quelle mesure n'est "fortement dominé" par aucun autre groupe). Si RG-SKY contient plus de K groupes, les top-K groupes sont renvoyés à l'utilisateur, puis le processus s'arrête. Sinon, on révisé le vecteur de paramètres de relaxation (le rendant plus permissif) et on recalcule le RG-SKY.

4.3.4.1 Algorithme naïf pour le calcul du RG-SKY

Cette section présente le premier algorithme (Algorithme 1) proposé pour implémenter l'approche RG-SKY. Il n'utilise aucune technique d'optimisation pour réduire l'espace de recherche.

Algorithm 1 RG-SKY naive algorithm

Result: Relaxed Group Skyline

Input : G-SKY : Skyline groups, \mathbb{G} : Groups, K : integer, γ : relaxation Vector

Output: RG-SKY

RG-SKY \leftarrow G-SKY //1ère étape est la génération de G-SKY

GetRestGroups(G-SKY, \mathbb{G});

SortbySkyPoints(G-REST); // phase de pré-traitement

DeleteZeroSkyGroups(G-REST); // phase de nettoyage

$j \leftarrow 1$;

while ($G-REST.length \leq k$) **do**

 // level : indique le nombre de points du Skyline de groupe actuel

 for($i=0; i < N; i++$) {

Compile μ_{MP} ($level[j].get(i), \gamma$); // $g' \in Rest, \mu_{MP_i}(g'i, gi)$

 } $j++$;

end

Return RG-SKY

4.3.4.2 Algorithme Salsa révisé

Cet algorithme est une version améliorée de l'algorithme naïf où l'on évite d'analyser tous les groupes générés pour extraire les groupes Skyline. Ce choix s'appuie sur l'étude comparative menée sur un ensemble d'algorithmes Skyline bien connus, comme expliqué ci-dessous.

Pour intégrer l'approche RG-SKY dans notre modèle du système d'aide à la décision proposé précédemment, sachant que le traitement des requêtes Skyline de groupe entraîne une procédure coûteuse, il est alors important de choisir le meilleur algorithme Skyline adaptatif à notre contexte. Le choix de Salsa (algorithme de tri et de limite Skyline) est justifié par le fait qu'il surmonte les principales limitations rencontrées dans les autres algorithmes généraux du Skyline, comme résumé dans le tableau 2.2 voir par exemple [PTFS03b, PTFS05b] présentés dans le chapitre 2.

Dans cette étude comparative, nous n'avons considéré que les algorithmes bien connus dans le domaine des requêtes Skyline.

L'algorithme Salsa utilisé dans l'évaluation de requêtes Skyline est une amélioration des algorithmes SFS et LESS (voir Tableau 2.2 du chapitre 2). Il s'efforce d'éviter d'analyser l'ensemble des données triées contrairement aux propositions précédentes, c'est le premier algorithme qui exploite les valeurs d'une fonction de notation monotone pour trier l'ensemble de données à lire et à comparer.

Salsa diffère des autres algorithmes génériques car il limite systématiquement le nombre de points lus et les tests de dominance. La conception de Salsa repose sur deux concepts clés : d'une part, une étape de tri des données d'entrée et, d'autre part, de choisir convenablement une fonction de tri. qui ne privilégie aucun attribut sur les autres (la fonction n'influence pas l'exactitude de Salsa mais seulement ses performances). Pour ces raisons, nous adaptons l'algorithme Salsa pour le calcul du Skyline de groupe et d'optimiser également le processus de relaxation de l'approche RG-SKY.

Pendant la phase de filtrage, l'algorithme lit et examine le reste des groupes. Chaque fois qu'un nouveau groupe est lu, il est comparé à la liste actuelle des groupes Skyline. Si un groupe domine le groupe courant, il est ignoré, sinon il est inséré dans la liste (en tant que skyline de groupe relaxé final) et l'algorithme vérifie sa condition de terminaison (Pstop). Si le seuil courant Pstop est inférieur ou égal à la fmin valeur du point, l'algorithme se termine et renvoie la liste complète du Skyline de groupe RG-SKY.

Cette condition de terminaison garantit qu'aucun groupe de données examiné ultérieurement ne devrait faire partie de la liste RG-SKY, l'algorithme évite ainsi d'analyser tout le reste de l'ensemble des données. Il existe de nombreuses fonctions limites dans la littérature, mais la fonction optimale qui peut limiter toute relation d'entrée plus que d'autres, est la fonction de coordonnées minimales (MCF) par rapport à Sum et Val présentés dans [BCP08b] (MCF est noté MinC, il trie d'abord les groupes compte tenu de la valeur de coordonnée minimale du groupe actuel et simultanément Sum la fonction des éléments du groupe est calculée et utilisée en cas d'égalité).

Algorithm 2 RG-SKY SALSA algorithm

Input : G-SKY : Group Skyline, \mathbb{G} : Groups, k : integer, γ : relaxation Vector, F : A monotone sorting function

Output: RG-SKY : Relaxed Group Skyline

RG-SKY \leftarrow G-SKY

GetRestGroups(G-SKY, \mathbb{G}); // G-Rest

SortbySkyPoints(G-Rest);

DeleteZeroSkyGroups(G-REST);

RG-SKY \leftarrow G-SKY; stop \leftarrow false; pstop \leftarrow undefined;

Sort G-REST according to F; // F(G-REST);

while (\neg pstop and G-REST $\neq \emptyset$) **do**

 G \leftarrow get next group from G-REST;

 G-REST \leftarrow G-REST \setminus {G}

if \neg MP(G, G-SKY) **then** RG-SKY \leftarrow RG-SKY \cup {G}, update pstop **then**

if pstop > G-SKY **then**

 stop \leftarrow true;

end

Return RG-SKY;

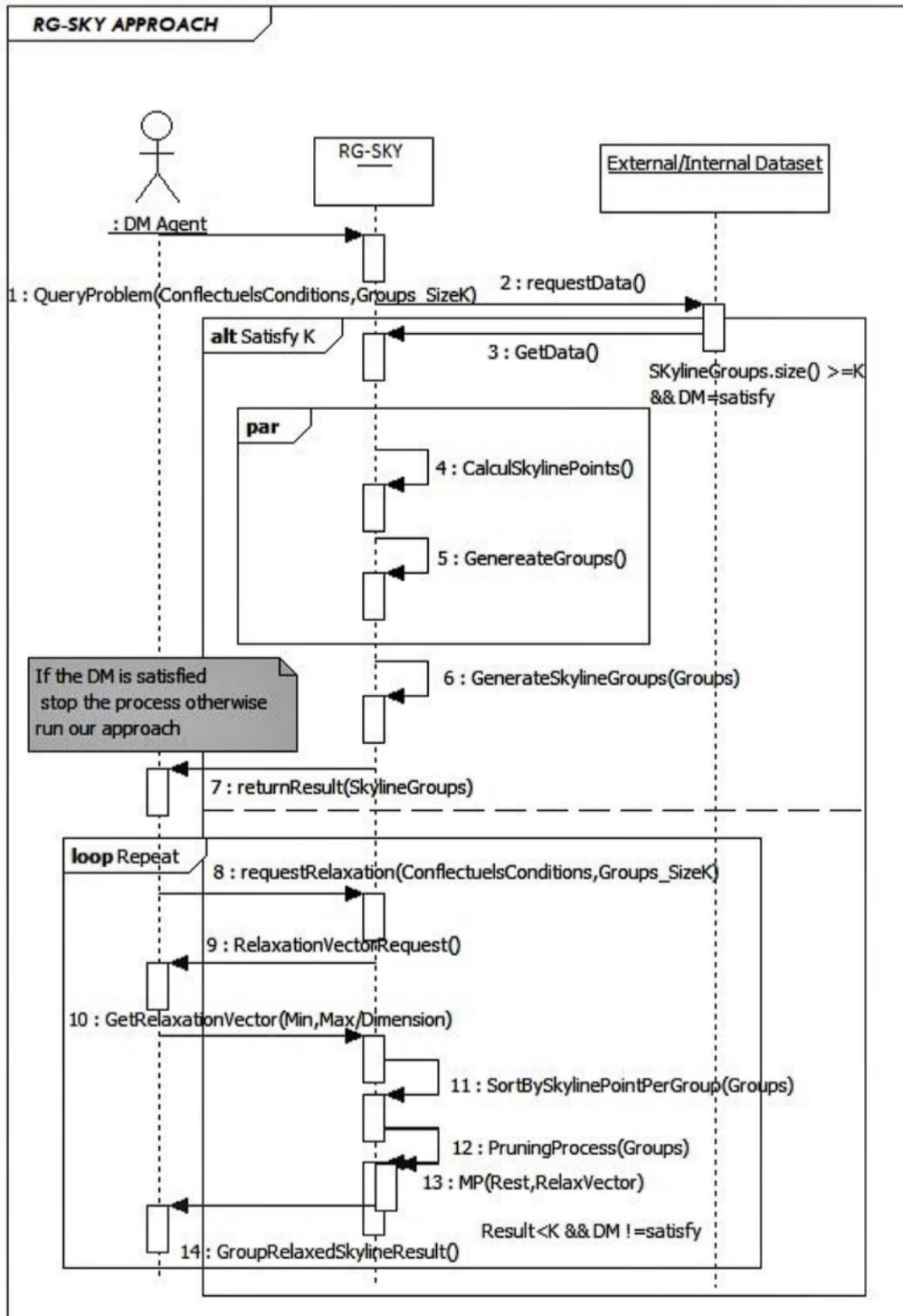


FIGURE 4.2

4.4 Étude expérimentale

Cette section présente l'étude expérimentale réalisée pour tester RG-SKY. Elle valide l'efficacité et la pertinence de l'approche RG-SKY pour relaxer les requêtes Skyline de groupe et mesure également quelques performances liées au temps de calcul.

4.4.1 Environnement expérimental

Les algorithmes sont testés sur une machine Dell Inc, modèle de système : Precision T1650 andruid dans un environnement Windows 10 Education 64 bits (10.0, Build 16299), en utilisant un processeur 3,4 GHz Intel(R) Core(TM) i7-3770 @ 3,40 GHz (8 processeurs) avec une mémoire principale de 8 Go de RAM en mode séquentiel (1 thread) et 500 Go de disque. Le benchmark de l'ensemble des données est généré à l'aide de la méthode décrite dans [BKS01b] suivant trois schémas de distribution classiques (corrélés, anti-corrélés et indépendante) et aussi "randataset". L'approche a été développée en Version des outils de modélisation Eclipse : Oxygen.3 Release (4.7.3), utilisant le langage Java.v9.

4.4.2 Tests expérimentaux

Les tests peuvent être classés en deux parties principales : Real Data (NBA, HOUSE, WEATHER) et des données synthétiques où nous modifions le type et la taille des données, les dimensions des groupes et des tuples et générons enfin des données à des fins de tests de vecteur de paramètre de relaxation.

Les études sur le Skyline existantes [IP12b, ZLH⁺14] utilisent le générateur de données proposé dans [BKS01b] avec la norme paramètres afin que les résultats puissent être directement comparés. Comme notre travail utilise le même genre de données, nous proposons de suivre la même méthodologie pour évaluer notre proposition. Les données génératrices dans Borzsonyi et al. (2001) génèrent des tuples (ou points) de base de données avec nombre de dimensions (ou attributs). Les tuples sont générés en utilisant l'une des trois distributions de valeurs : – corrélés (corr) : tuples, qui sont bons dans une dimension et ont tendance à être bons dans d'autres dimensions également ; – anti-corrélés (anti) : tuples, qui sont bons dans une dimension et sont mauvais dans au moins une autre dimension ; – indépendant (indep) : tuples sont générés en utilisant une distribution uniforme.

Nous incluons également trois jeux de données réels (voir le tableau 4.2 pour la spécification de ces jeux de données) qui sont couramment utilisés pour évaluer les algorithmes du Skyline [CŠAB15] :

- NBA (les statistiques des joueurs de basket pendant les saisons régulières),
- HOUSE (argent dépensé en un an par une famille américaine pour six types de dépenses différentes),
- WEATHER (Moyens des précipitations totales mensuelles et l'altitude à plus d'un demi-million d'emplacements de capteurs).

Données	Cardinalité	Dimension
NBA	17,264	8
House	127,931	6
Weather	566,268	15

TABEAU 4.2 – Spécifications des jeux de données réelles

Paramètres	Valeurs	Valeur par défaut
Groupes [N]	NBA = { 500 }, HOUSE = { 5911 }, WEATHER = { 4438 }, Corrélée = { 10, 50, 100, 500, 2000-3612281 }, Anti-corrélée = { 2700 }, Indépendant = { 1500 }	10, 50, 100, 500, 1000
Dataset distribution schema	Corrélée, Indépendant, anti-corrélée	Corrélée
Dimension de groupe [l]	2, 3, 4, 5	2
Vecteur de relaxation [(γ_1, γ_2)]	$\gamma_1 = [0, 1[$, $\gamma_2 = [0, 1[$	(0, 1)

TABLEAU 4.3 – Ensemble de paramètres (où N représente le nombre de groupes d'entrée)

- Enfin, un dernier test est effectué sur des données réelles générées à l'aide du générateur Skyline (randdataset).

Nous résumons nos valeurs d'entrée (constante/variable) dans le tableau 4.3 et nos tests expérimentaux dans la figure 4.3.

4.4.3 Résultats expérimentaux

- Cas 1 : données réelles

La figure 4.4 montre le nombre de groupes Skyline relaxés dans un type de données différent et le temps d'exécution de l'approche RG-SKY. RG-SKY fournit plus de groupes par rapport à l'ensemble G-SKY. On peut observer que pour les trois jeux de données (NBA, WEATHER, HOUSE) G-SKY ne contient qu'un seul groupe.

Le temps d'exécution décrit dans la figure 4.4 (résultant de l'algorithme 1) n'est pas similaire pour les trois jeux de données en raison de leurs corrélations et tailles différentes, alors que ce temps est similaire pour le jeu de données NBA (corrélé).

Après la première exécution, nous proposons un algorithme optimisé (Algorithme 2) qui conduit à 97,20 % d'amélioration de la version naïve (algorithme 1).

- Cas 2 : Données synthétiques (corrélées)

- Variation du type de date : La figure 4.5 montre le temps d'exécution et le nombre renvoyé de groupes Skyline aux détenteurs dans différents types de données : données corrélées, anti-corrélées et indépendantes. Comme on peut le voir, l'ensemble G-SKY contient toujours un groupe pour les trois ensembles de données par rapport à RG-Sky qui renvoie plus d'un groupe, à l'exception des données corrélées qui renvoient le même nombre de groupes Skyline.

D'autre part, on peut noter que les données indépendantes et anti-corrélées prennent du temps par rapport aux données corrélées. Nous notons également que notre algorithme RG-SKY Salsa (Algorithme 2) prend autant ou moins de temps par rapport à l'algorithme naïf G-SKY existant (Sauf pour les données anti-corrélées).

C'est pourquoi nous décidons de poursuivre nos expérimentations uniquement sur des données corrélées.

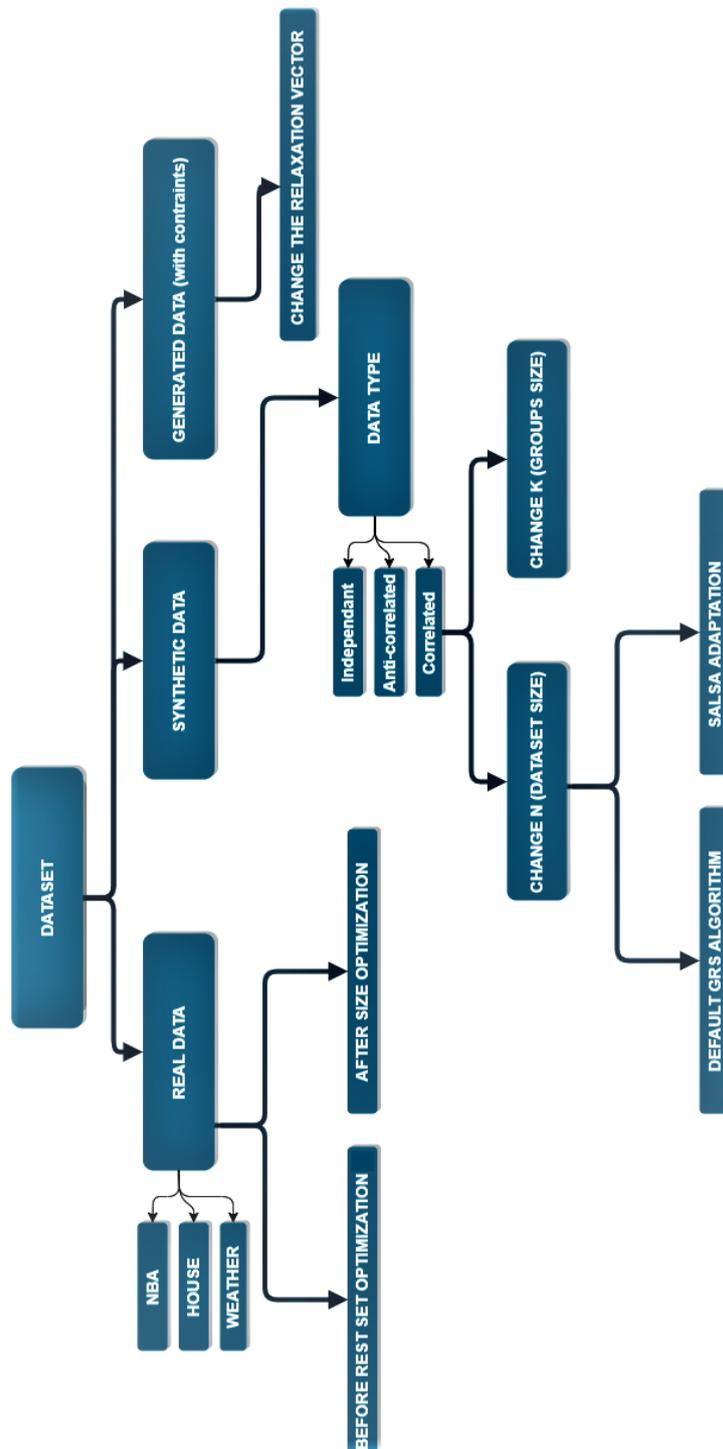


FIGURE 4.3 – Résumé des différents tests expérimentaux

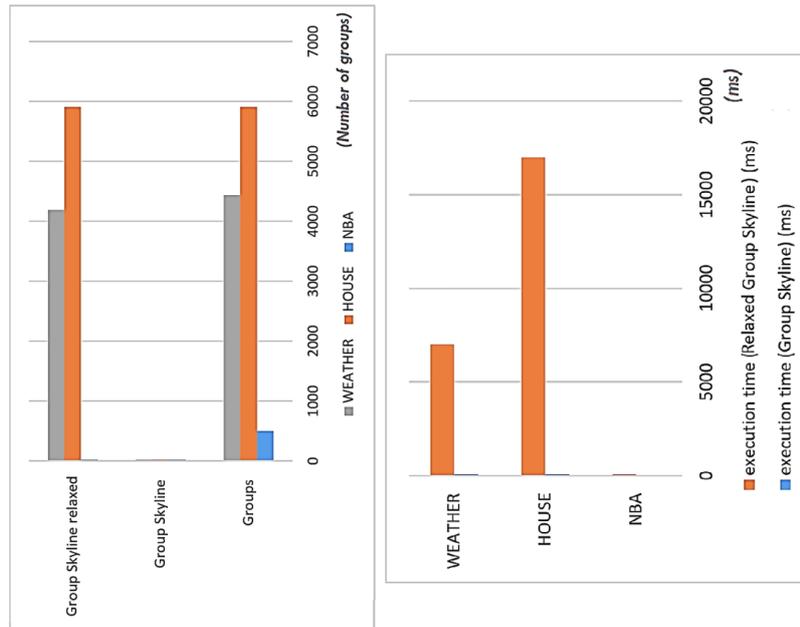


FIGURE 4.4 – Temps d’exécution des données réelles et nombre de groupes Skyline relaxés renvoyés

– Variation de la taille des données : La figure 4.6 montre que l’approche RG-SKY est très efficace. Nous obtenons plus de groupes Skyline avec une consommation de temps presque similaire lorsque les groupes d’entrée ne dépassent pas 500 groupes.

– Variation de la dimension du groupe

Dans cette partie, nous nous concentrons sur la taille des dimensions et nous faisons varier les dimensions du groupe (c’est-à-dire le paramètre l) au lieu du nombre de groupes d’entrée (c’est-à-dire N).

La figure 4.7 montre que le temps d’exécution d’approche RG-SKY est acceptable lorsque le nombre d’éléments (c’est à dire l) dans le groupe ne dépasse pas 4. Par exemple, pour 100 tuples, si $l=3$ alors le temps d’exécution pour la génération du groupe : G - Le calcul SKY et le calcul RG-SKY sont respectivement de 4754 (ms), 7 (ms), 19 (ms). Alors que pour $l=5$, on obtient respectivement 3612281 (ms), 3022 (ms), 242000 (ms).

— Cas 3 : Données générées

Pour le dernier test, nous générons des groupes en utilisant des valeurs de données normalisées afin d’analyser l’impact des valeurs du vecteur de relaxation sur l’approche RG-SKY. En raison de la normalisation des données, nous utilisons la même relation préférée $MPG_i(1,2)$ pour tous les attributs Skyline i . La figure 4.8 montre deux cas pour la relation $MPG_i(\gamma_1, \gamma_2)$:

En Cas 1 : γ_1 est fixe ($\gamma_1=0$) et γ_2 varie pour augmenter la zone de relaxation, le résultat obtenu montre que la taille de Skyline de groupe relaxé augmente lorsque γ_2 est plus grand mais le temps d’exécution reste acceptable.

En Cas 2 : γ_1 et γ_2 changent tous les deux, le résultat obtenu montre que la taille de Skyline de groupe détendu augmente de la même manière que dans le premier cas. On peut observer que le temps d’exécution peut être considéré comme raisonnable par rapport aux nombres de groupes Skyline générés relaxés.

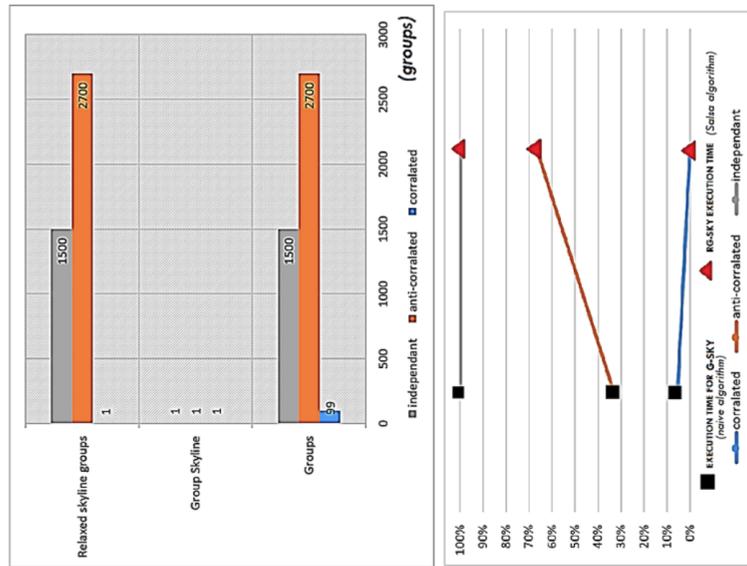


FIGURE 4.5 – Temps d'exécution des données synthétiques et nombre de groupes skyline détendus dans différents types de données

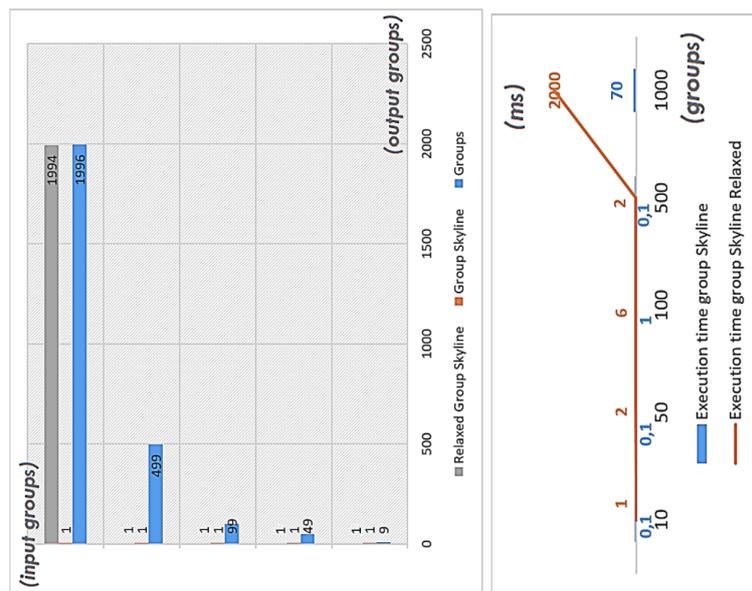


FIGURE 4.6 – Synthetic correlated data with different dataset size

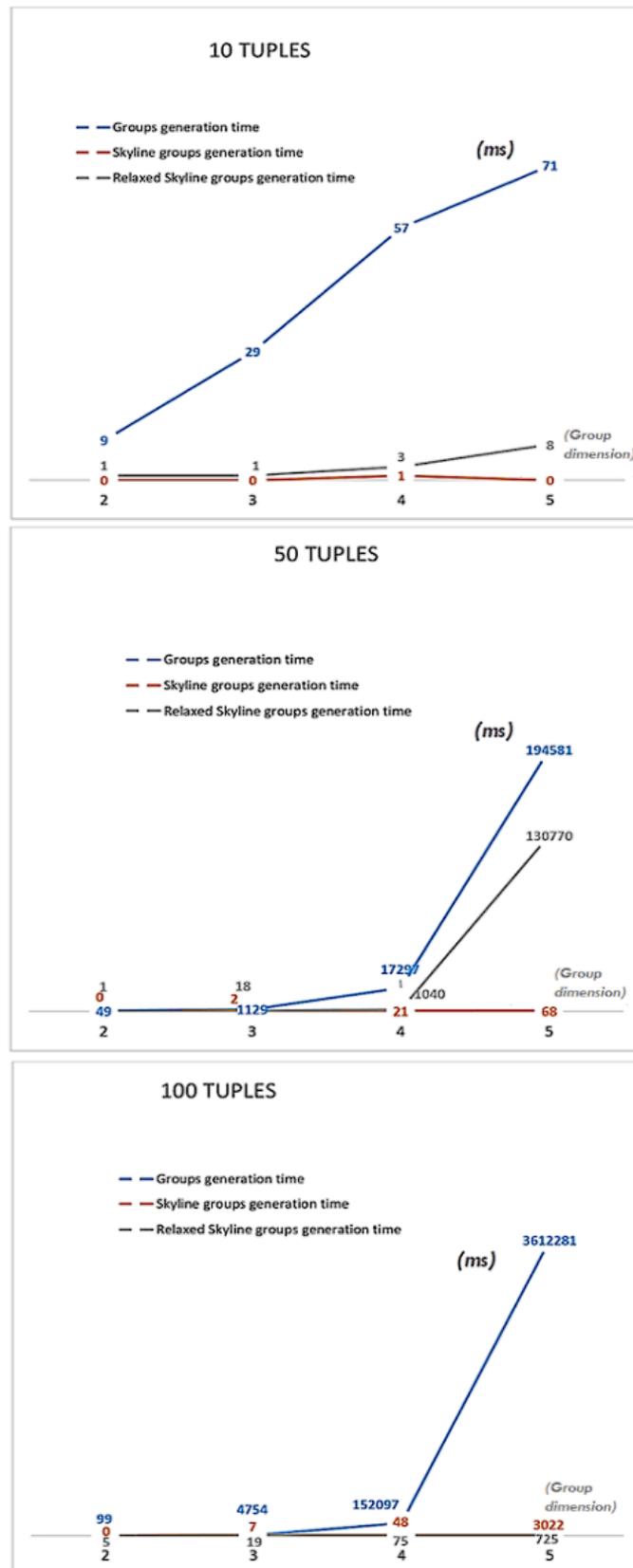


FIGURE 4.7 – données synthétiques corrélées avec des valeurs dynamiques du groupe K (dimensions) et du groupe N (nombre de groupes)

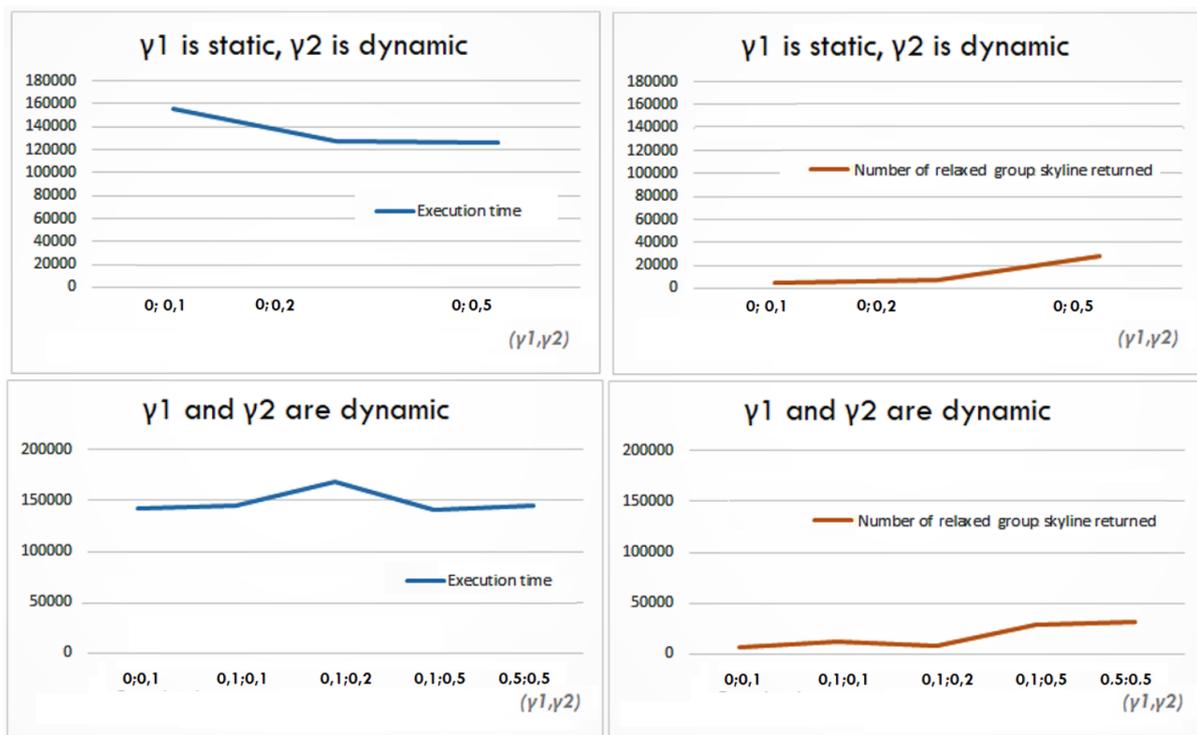


FIGURE 4.8 – données synthétiques corrélées avec des valeurs dynamiques du groupe K (dimensions) et du groupe N (nombre de groupes)

4.5 Conclusion et Bilan

Dans ce chapitre, nous avons présenté une approche de relaxation pour les requêtes Skyline de groupes. Elle permet d'élargir le Skyline de groupe avec des groupes intéressants d'une manière contrôlée, et facilite ainsi la prise de décision. De plus, pour mieux répondre aux besoins et attentes des décideurs, l'approche RG-SKY leur offre le choix de la relaxation appropriée grâce aux différents paramètres d'entrée. Le concept clé de cette approche est une relation floue particulière, nommée "beaucoup préférée", dont la sémantique est définie par l'utilisateur.

De plus, deux algorithmes pour calculer le Skyline de groupe relaxé sont proposés, le premier est une version naïve de la méthode RG-SKY et le second est une méthode optimisée. Cette version utilise l'algorithme Salsa qui est utilisé pour la première fois pour extraire les groupes à la place de points individuels dans le cadre d'une relaxation. L'étude expérimentale montre que l'approche RG SKY est une bonne alternative en terme de temps d'exécution, nombre de groupes Skyline et de satisfaction des utilisateurs.



Conclusion et perspectives

Conclusion

Dans le cadre du travail mené dans cette thèse, nous avons contribué à la construction d'un nouveau modèle de prise de décision distribuée et collective en combinant la technologie agent et le paradigme Skyline. Ce qui permet d'améliorer substantiellement le processus de prise de décision en termes d'optimisation, de fiabilité et d'intelligence.

Dans la première partie de ce manuscrit, nous avons vu l'évolution des SI qui a donné naissance aux systèmes de prise de décision (DSS). Puis, les systèmes DSS ont conduit aux systèmes DDSS en prenant en compte les données et utilisateurs localisées sur des sites géographiquement distants. Nous avons aussi présenté une étude comparative des méthodologies existantes qui modélisent les DDSS par le biais des systèmes multi-agents et nous avons pu identifier un ensemble de problèmes, notamment l'absence de mécanisme de tolérance aux fautes et l'absence de vérification formelle que nous avons présentés en détail dans le chapitre 3 où nous avons proposé une modélisation des DDSS, en utilisant l'approche MAS, afin de pallier aux faiblesses des modélisations existantes.

La deuxième partie de notre contribution a trait à l'amélioration de la prise de décision en exploitant les données présentes dans le système DDSS. Elle consiste à intégrer l'approche Skyline classique (individuel) dans le processus interne du DDSS. L'algorithme Salsa, utilisé sur les tables de décisions, rend le calcul du Skyline plus performant et ainsi gagner en termes de temps de réponse pour la prise de décision.

Dans la troisième partie de notre contribution, nous nous sommes intéressés à une variante particulière du Skyline, dite Skyline de groupe. Malgré les différents avantages apportés par cette variante dans la prise de décisions, elle souffre de certaines insuffisances majeures qui pourraient conduire à des situations d'échec (par manque d'éléments de Skyline). Pour traiter ce problème critique, nous avons proposé une méthode de relaxation du Skyline de groupe calculé. Cette relaxation est fondée sur une relation de préférences floues définie par l'utilisateur. Un contrôle des résultats du Skyline est ainsi mis en place dans le processus de prise de décision. Des algorithmes efficaces et optimisés ont été développés pour le calcul de la variante relaxée du Skyline de groupe en exploitant certaines propriétés sémantiques de la définition de la relaxation proposée.

Toutes nos propositions ont été validées en menant un ensemble d'expérimentations sur des données réelles et synthétiques présentées en chapitres 3 et 4.

Perspectives

L'ensemble des contributions proposées dans le cadre de cette thèse ouvre la voie vers de nouveaux champs d'études permettant ainsi la poursuite de nos travaux de recherche. Les perspectives détaillées dans ce qui suit s'articulent autour de trois axes principaux :

Pour notre première contribution, nous envisageons examiner une modélisation fondée sur les bi-graphes pour le modèle DDSS, à base de systèmes multi-agents, proposé. Cette nouvelle modélisation faciliterait la mise en place d'une validation formellement du système.

Pour la deuxième partie de notre contribution, nous comptons implémenter les algorithmes proches de l'algorithme Salsa afin de généraliser notre proposition sur l'ensemble des algorithmes Skyline individuels existants.

Pour notre troisième contribution, nous comptons améliorer les performances en prenant en considération la partie génération des groupes en éliminant certains groupes générés en utilisant des méthodes de hiérarchie basées sur le nombre de points Skyline par groupe. Nous utiliserons l'algorithme optimisé que nous avons proposé pour le Skyline individuel (chapitre 3) afin de l'adapter à notre algorithme 2 de Salsa (l'original) présenté dans le dernier chapitre.

Une autre voie de nos futurs travaux vise à explorer la possibilité d'utiliser l'approche RG-SKY pour étendre la définition 2 du chapitre 2 du Skyline de groupe basée sur la permutation en utilisant la représentation par graphes. Nous envisageons également d'explorer le calcul parallèle pour la génération de relaxation progressive du Skyline de groupe afin de réduire la consommation de temps de l'approche. Enfin, nous essayons de générer automatiquement le vecteur de relaxation en tenant compte des préférences sur attributs données par l'utilisateur.

Bibliographie

- [A⁺09] Sohail Asghar et al. A survey on multi-criteria decision making approaches. In *2009 International conference on emerging technologies*, pages 321–325. IEEE, 2009. (Cité en page 53)
- [ADLMGO12] Pablo García Ansola, Javier De Las Morenas, Andres Garcia, and Javier Otamendi. Distributed decision support system for airport ground handling management using wsn and mas. *Engineering Applications of Artificial Intelligence*, 25(3) :544–553, 2012. (Cité en page 24)
- [Ale02] Larry Alexander. Decision support systems in the 21st century. *ACM SIGSOFT Software Engineering Notes*, 27(5) :104–104, 2002. (Cité en page ix), (Cité en page 15), (Cité en page 17)
- [ANOM12a] Abdelkader Adla, Bakhta Nached, and Abdelkader Ould-Mahraz. Multi-agents model for web-based collaborative decision support systems. In *ICWIT*, pages 294–299, 2012. (Cité en page 23)
- [ANOM12b] Abdelkader Adla, Bakhta Nached, and Abdelkader Ould-Mahraz. Multi-agents model for web-based collaborative decision support systems. In *ICWIT*, pages 294–299. Citeseer, 2012. (Cité en page 24), (Cité en page 46)
- [ANOW16] C Aura, N Ngisiange, G Okeyo, and N Wambiji. Multi-agent systems and distributed constraint satisfaction for decision support in marine ecosystem management. 2016. (Cité en page 24)
- [ASL04a] Wen-Guo Ai, Jie Sun, and Hui Li. A distributed marketing decision support system based on multi-intelligent-agent. In *Proceedings of 2004 international conference on machine learning and cybernetics (IEEE Cat. No. 04EX826)*, volume 1, pages 233–238. IEEE, 2004. (Cité en page 24)
- [ASL04b] Wen-Guo Ai, Jie Sun, and Hui Li. A distributed marketing decision support system based on multi-intelligent-agent. In *Proceedings of ICMLC*, volume 1, pages 233–238. IEEE, 2004. (Cité en page 24), (Cité en page 46)
- [BCP06] Iliaria Bartolini, Paolo Ciaccia, and Marco Patella. Salsa : Computing the skyline without scanning the whole sky. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 405–414, 2006. (Cité en page 53), (Cité en page 55)
- [BCP08a] Iliaria Bartolini, Paolo Ciaccia, and Marco Patella. Efficient sort-based skyline evaluation. *ACM Transactions on Database Systems (TODS)*, 33(4) :1–49, 2008. (Cité en page 53), (Cité en page 55)
- [BCP08b] Iliaria Bartolini, Paolo Ciaccia, and Marco Patella. Efficient sort-based skyline evaluation. *ACM Trans. Database Syst.*, 33 :31 :1–31 :49, 2008. (Cité en page 72)

- [BdNdL16] Jean-Pierre Briot, Nathalia Moraes de Nascimento, and Carlos José Pereira de Lucena. A multi-agent architecture for quantified fruits : Design and experience. In *28th International Conference on Software Engineering & Knowledge Engineering (SEKE'2016)*, pages 369–374. SEKE/Knowledge Systems Institute, PA, USA, 2016. (Cité en page 21), (Cité en page 45)
- [BKS01a] Stephan Borzsony, Donald Kossmann, and Konrad Stocker. The skyline operator. In *Proceedings 17th international conference on data engineering*, pages 421–430. IEEE, 2001. (Cité en page 29), (Cité en page 65)
- [BKS01b] Stephan Borzsony, Donald Kossmann, and Konrad Stocker. The skyline operator. In *proc. 17th Inter. Conf. on data engineering*, pages 421–430. IEEE, 2001. (Cité en page 31), (Cité en page 53), (Cité en page 74)
- [Bri09] Jean-Pierre Briot. Composants logiciels et systemes multi-agents, 2009. (Cité en page 20)
- [CD18] Xiaowei Cui and Leigang Dong. An efficient algorithm to compute compositional skyline. In *IOP Conf. Series : Materials Science and Engineering*, volume 466, page 012021. IOP Publishing, 2018. (Cité en page 36), (Cité en page 37)
- [CGGL03] Jan Chomicki, Parke Godfrey, Jarek Gryz, and Dongming Liang. Skyline with presorting. In *ICDE*, volume 3, pages 717–719, 2003. (Cité en page 29)
- [CŠAB15] Sean Chester, Darius Šidlauskas, Ira Assent, and Kenneth S Bøgh. Scalable parallelization of skyline computation for multi-core processors. In *2015 IEEE 31st Inter. Conf. on Data Engineering*, pages 1083–1094. IEEE, 2015. (Cité en page 74)
- [CSCZ05] Jia-Yu Chi, Ling Sun, Xue-Guang Chen, and Jin-Long Zhang. Architecture and design of distributed decision support system based on agent grid. In *2005 International Conference on Machine Learning and Cybernetics*, volume 1, pages 321–327. IEEE, 2005. (Cité en page 23)
- [CSL13] Yu-Chi Chung, I-Fang Su, and Chiang Lee. Efficient computation of combinatorial skyline queries. *Information Systems*, 38(3) :369–387, 2013. (Cité en page 36), (Cité en page 37)
- [DKMP00] Didier Dubois, Etienne Kerre, Radko Mesiar, and Henri Prade. Fuzzy interval analysis. In *Fundamentals of fuzzy sets*, pages 483–581. Springer, 2000. (Cité en page 70)
- [DLCL18a] Leigang Dong, Guohua Liu, Xiaowei Cui, and Tianyu Li. Finding group-based skyline over a data stream in the sensor network. *Information*, 9(2) :33, 2018. (Cité en page 36), (Cité en page 37)
- [DLCL18b] Leigang Dong, Guohua Liu, Xiaowei Cui, and Tianyu Li. Finding group-based skyline over a data stream in the sensor network. *Information*, 9(2) :33, 2018. (Cité en page 36)
- [DN05] Maghri Dorra and Amara Nejla. Mobile agent integration in a hierarchical multi-agent system - intégration d'un agent mobile dans un système multi-agent hiérarchique. In *MajecSTIC*, pages 426–429, 2005. (Cité en page 48)
- [DS04a] Claude Duvallat and Bruno Sadeg. Anytime multi-agent systems for the design of decision support systems - des systèmes multi-agents anytime pour la

-
- conception de systèmes d'aide à la décision. *Revue des Sciences et Technologies de l'Information-Série TSI*, 23(8) :997–1025, 2004. (Cité en page 24), (Cité en page 46), (Cité en page 47)
- [DS04b] Claude Duvallet and Bruno Sadeg. Des systèmes multi-agents anytime pour la conception de systèmes d'aide à la décision. *Revue des Sciences et Technologies de l'Information-Série TSI : Technique et Science Informatiques*, 23(8) :997–1025, 2004. (Cité en page 23)
- [EL90] Hyun B Eom and Sang M Lee. A survey of decision support system applications (1971–april 1988). *Interfaces*, 20(3) :65–79, 1990. (Cité en page 10)
- [EMF] EMF. Eclipse modeling framework (emf). <https://www.eclipse.org/modeling/emf/>. Online; accessed 21 March 2014. (Cité en page 52)
- [Fer97a] Jacques Ferber. *Multi-Agent Systems : towards intelligence - Les systèmes multi-agents : vers une intelligence collective*. InterEditions, 1997. (Cité en page 48)
- [Fer97b] Nils Ferrand. *Modèles Multi-Agents pour l'aide à la décision et la négociation en aménagement du territoire*. PhD thesis, Université Joseph-Fourier-Grenoble I, 1997. (Cité en page 23)
- [FMGG09] Guisseppi Forgionne, Manuel Mora, Jatinder ND Gupta, and Ovsei Gelman. Decision-making support systems. In *Encyclopedia of Information Science and Technology, Second Edition*, pages 978–984. IGI Global, 2009. (Cité en page 53)
- [FW99] Jacques Ferber and Gerhard Weiss. *Multi-agent systems : an introduction to distributed artificial intelligence*, volume 1. Addison-wesley Reading, 1999. (Cité en page 17), (Cité en page 19), (Cité en page 20)
- [Gau07] Nicolas A Gaud. *Systèmes multi-agents holoniques : de l'analyse à l'implantation : méta-modèle, méthodologie, et simulation multi-niveaux*. PhD thesis, Besançon, 2007. (Cité en page 20)
- [GLW⁺16a] Xi Guo, Hailing Li, Aziguli Wulamu, Yonghong Xie, and Yajing Fu. Efficient processing of skyline group queries over a data stream. *Tsinghua Science and Technology*, 21(1) :29–39, 2016. (Cité en page 36), (Cité en page 37)
- [GLW⁺16b] Xi Guo, Hailing Li, Aziguli Wulamu, Yonghong Xie, and Yajing Fu. Efficient processing of skyline group queries over a data stream. *Tsinghua Science and Technology*, 21(1) :29–39, 2016. (Cité en page 38)
- [GSG⁺05] Parke Godfrey, Ryan Shipley, Jarek Gryz, et al. Maximal vector computation in large data sets. In *VLDB*, volume 5, pages 229–240, 2005. (Cité en page 29)
- [GSM71] George Anthony Gorry and Michael S Scott Morton. A framework for management information systems. 1971. (Cité en page 11)
- [GT07] Marlene Goncalves and Leonid Tineo. Fuzzy dominance skyline queries. In *Inter. Conf. on Database and Expert Systems Applications*, pages 469–478. Springer, 2007. (Cité en page 31)
- [GVMA⁺06a] H Gonzalez-Velz, M Mier, Carles Arus, Bernardo Celda, Sabine Van Huffel, Paul Lewis, Andrew Peet, and M Robles. Agent-based distributed decision support system for brain tumour diagnosis and prognosis. 2006. (Cité en page 24)

- [GVMA⁺06b] H Gonzalez-Velz, M Mier, Carles Arus, Bernardo Celda, Sabine Van Huffel, Paul Lewis, Andrew Peet, and M Robles. Agent-based distributed decision support system for brain tumour diagnosis and prognosis. 2006. (Cité en page 24), (Cité en page 46)
- [GXI12] Xi Guo, Chuan Xiao, and Yoshiharu Ishikawa. Combination skyline queries. In *Transactions on Large-Scale Data-and Knowledge-Centered Systems VI*, pages 1–30. Springer, 2012. (Cité en page 36), (Cité en page 37)
- [HAF09] MH HAFFAF. *Un Système Interactif d’Aide à la Décision de Groupe En Aménagement du Territoire*. PhD thesis, Université d’Oran, 2009. (Cité en page 17)
- [HJW94] CW Holsapple, VS Jacob, and AB Whinston. Model management. *Operations Research and Artificial Intelligence*, page 22, 1994. (Cité en page 9), (Cité en page 10)
- [IP12a] Hyeonseung Im and Sungwoo Park. Group skyline computation. *Information Sciences*, 188 :151–169, 2012. (Cité en page 29), (Cité en page 38)
- [IP12b] Hyeonseung Im and Sungwoo Park. Group skyline computation. *Information Sciences*, 188 :151–169, 2012. (Cité en page 36), (Cité en page 37), (Cité en page 69), (Cité en page 74)
- [JZL⁺15] Tao Jiang, Bin Zhang, Dan Lin, Yunjun Gao, and Qing Li. Incremental evaluation of top-k combinatorial metric skyline query. *Knowledge-Based Systems*, 74 :89–105, 2015. (Cité en page 29), (Cité en page 35), (Cité en page 37)
- [Kee80] Peter GW Keen. Adaptive design for decision support systems. *Acm Sigoa Newsletter*, 1(4-5) :15–25, 1980. (Cité en page ix), (Cité en page 10), (Cité en page 12)
- [Ken03] Stuart Kent. Model driven language engineering. *Electr. Notes Theor. Comput. Sci.*, 72(4) :6, 2003. (Cité en page 52)
- [Kho08] Smaïl Khouider. *Outils d’aide à la décision pour la prise de commandes imprévues*. PhD thesis, Université Paul Verlaine-Metz, 2008. (Cité en page 24)
- [KK02] Werner Kießling and Gerhard Köstler. Preference sql—design, implementation, experiences. In *VLDB’02 : Proceedings of the 28th International Conference on Very Large Databases*, pages 990–1001. Elsevier, 2002. (Cité en page 53)
- [KLP75] Hsiang-Tsung Kung, Fabrizio Luccio, and Franco P Preparata. On finding the maxima of a set of vectors. *Journal of the ACM (JACM)*, 22(4) :469–476, 1975. (Cité en page 35), (Cité en page 53)
- [KRR02] Donald Kossmann, Frank Ramsak, and Steffen Rost. Shooting stars in the sky : An online algorithm for skyline queries. In *VLDB’02 : Proceedings of the 28th International Conference on Very Large Databases*, pages 275–286. Elsevier, 2002. (Cité en page 29)
- [KT99] H Kivijarvi and Markku Tuominen. Computer based intelligence, design, choice, implementation, and control of intangible investments projects. In *Systems Sciences, proceedings of the 32nd Annual HICSS-32. Hawaii*, pages 11–pp. IEEE, 1999. (Cité en page 47), (Cité en page 48)
- [KT17] Christos Kalyvas and Theodoros Tzouramanis. A survey of skyline query processing. *CoRR*, abs/1704.01788, 2017. (Cité en page 55)

-
- [LAEM⁺07] Magí Lluch-Ariet, Francesc Estanyol, Mariola Mier, Carla Delgado, Horacio González-Vélez, Tiphaine Dalmas, Montserrat Robles, Carlos Sáez, Javier Vicente, Sabine Van Huffel, et al. On the implementation of healthagents : agent-based brain tumour diagnosis. In *Agent Technology and e-Health*, pages 5–24. Springer, 2007. (Cité en page 24), (Cité en page 46)
- [Lai02] Hakim Laichour. *Multi-Agent Modeling and Decision Support : Application to Correspondence Regulation in Urban Transport Networks - Modélisation multi-agent et aide à la décision : application à la régulation des correspondances dans les réseaux de transport urbain*. PhD thesis, Lille 1, 2002. (Cité en page 48)
- [LB13] Christoph Lofi and Wolf-Tilo Balke. On skyline queries and how to choose from pareto sets. In *Advanced Query Processing*, pages 15–36. Springer, 2013. (Cité en page 53)
- [LD12] DE MAGISTER LE DIPLOME. *MODELE MULTI-AGENTS POUR L'AIDE A LA DECISION DE GROUPE*. PhD thesis, Université d'Oran, 2012. (Cité en page 17)
- [LLH17] Ming-Yen Lin, Yueh-Lin Lin, and Sue-Chen Hsueh. Discovering group skylines with constraints by early candidate pruning. In *Inter. Conf. on Advanced Data Mining and Appl.*, pages 49–62. Springer, 2017. (Cité en page 36), (Cité en page 37)
- [LXP⁺15a] Jinfei Liu, Li Xiong, Jian Pei, Jun Luo, and Haoyu Zhang. Finding pareto optimal groups : Group-based skyline. *Proceedings of the VLDB Endowment*, 8(13) :2086–2097, 2015. (Cité en page 32), (Cité en page 36)
- [LXP⁺15b] Jinfei Liu, Li Xiong, Jian Pei, Jun Luo, and Haoyu Zhang. Finding pareto optimal groups : Group-based skyline. *proc. of the VLDB Endowment*, 8 :2086–2097, 2015. (Cité en page 35), (Cité en page 37)
- [LYX⁺18] Kenli Li, Zhibang Yang, Guoqing Xiao, Keqin Li, et al. Progressive approaches for pareto optimal groups computation. *IEEE Transactions on Knowledge and Data Engineering*, 31(3) :521–534, 2018. (Cité en page 36), (Cité en page 37)
- [LZH⁺12] Chengkai Li, Nan Zhang, Naeemul Hassan, Sundaresan Rajasekaran, and Gautam Das. On skyline groups. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2119–2123, 2012. (Cité en page 36)
- [MBA16a] Abdelkader Ould Mahraz, Djamila Bouhalouan, and Abdelkader Adla. Facilitating virtual group decision making. *Procedia Computer Science*, 83 :1050–1055, 2016. (Cité en page 23)
- [MBA16b] Abdelkader Ould Mahraz, Djamila Bouhalouan, and Abdelkader Adla. Facilitating virtual group decision making. *Procedia Computer Science*, 83 :1050–1055, 2016. (Cité en page 24), (Cité en page 46)
- [MOF] MOF. Mof official documentation. <http://www.omg.org/mof/>. Online ; accessed 21 March 2014. (Cité en page 52)
- [NA14] Bakhta Nachet and Abdelkader Adla. An agent-based distributed collaborative decision support system. *Intelligent Decision Technologies*, 8(1) :15–34, 2014. (Cité en page ix), (Cité en page 17), (Cité en page 21), (Cité en page 45)

- [Nfa08] El Habib Nfaoui. Architecture d'aide à la décision distribuée et de simulation proactive dans les chaînes logistiques : Une approche multi agent. 2008. (Cité en page 24)
- [NHS18] Sana Nadouri, Allel Hadjali, and Zaïdi Sahnoun. Group skyline computation : An overview. In *proc. of the 36th Computer Workshop of Organizations and Information Systems and Business Intelligence Decision Making, Big Data and Data Science, INFORSID, France, May 28-31, 2018.*, 2018. (Cité en page 36)
- [NOSH18] Sana Nadouri, Yassine Ouhammou, Zaidi Sahnoun, and Allel Hadjali. Towards a multi-agent approach for distributed decision support systems. In *2018 IEEE 27th Inter. Conf. on Enabling Technologies : Infrastructure for Collaborative Enterprises (WETICE)*, pages 72–77. IEEE, 2018. (Cité en page 70)
- [NSH18] Sana Nadouri, Zaïdi Sahnoun, and Allel Hadjali. Using g-skyline to improve decision-making. In *proc. of the 3rd Inter. Conf. on Advanced Aspects of Software Engineering, ICAASE 2018, Constantine, Algeria, December 1-2, 2018.*, pages 141–150, 2018. (Cité en page 36)
- [OCL] OCL. The object constraint language specification version 2.4. <https://www.omg.org/spec/OCL/2.4/>. Online ; accessed 21 March 2014. (Cité en page 52)
- [OHB08] Sarah Oufella, Djamila Hamdadou, and Karim Bouamrane. Proposition d'un modèle d'aide à la négociation pour les problèmes d'aménagement du territoire. In *Actes du Cinquième Colloque sur l'Optimisation et les Systèmes d'Information COSI'2008*, page 393, 2008. (Cité en page 17)
- [OHB09] Sarah Oufella, Djamila Hamdadou, and Karim Bouamrane. Conception d'un système interactif d'aide à la décision collective en localisation spatiale. In *CIIA*, 2009. (Cité en page 23)
- [PE15] Timotheus Preisinger and Markus Endres. Looking for the best, but not too many of them : Multi-level and top-k skylines. 2015. (Cité en page 53)
- [Pot10] Katia Potiron. *Systèmes multi-agents et tolérance aux fautes : conséquences de l'autonomie des agents*. PhD thesis, Paris 6, 2010. (Cité en page 20)
- [PR98] Patrice Perny and Marc Roubens. Fuzzy preference modeling. In *Fuzzy sets in decision analysis, operations research and statistics*, pages 3–30. Springer, 1998. (Cité en page 69), (Cité en page 70)
- [PTFS03a] Dimitris Papadias, Yufei Tao, Greg Fu, and Bernhard Seeger. An optimal and progressive algorithm for skyline queries. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 467–478, 2003. (Cité en page 29)
- [PTFS03b] Dimitris Papadias, Yufei Tao, Greg Fu, and Bernhard Seeger. An optimal and progressive algorithm for skyline queries. In *ACM SIGMOD Inter. Conf. on Management of Data, SIGMOD '03*, pages 467–478, New York, NY, USA, 2003. ACM. (Cité en page 71)
- [PTFS05a] Dimitris Papadias, Yufei Tao, Greg Fu, and Bernhard Seeger. Progressive skyline computation in database systems. *ACM Transactions on Database Systems (TODS)*, 30(1) :41–82, 2005. (Cité en page 29)

-
- [PTFS05b] Dimitris Papadias, Yufei Tao, Greg Fu, and Bernhard Seeger. Progressive skyline computation in database systems. *ACM Trans. Database Syst.*, 30(1) :41–82, March 2005. (Cité en page 71)
- [PW16] Peng Peng and Raymond CW Wong. Skyline queries and pareto optimality. In *Encyclopedia of database systems*, page 7. 2016. (Cité en page 53)
- [RCBV15] Alessandro Ricci, Angelo Croatti, Pietro Brunetti, and Mirko Viroli. Programming mirror worlds : An agent-oriented programming perspective. In *International Workshop on Engineering Multi-Agent Systems*, pages 191–211. Springer, 2015. (Cité en page 48)
- [Rod11] Blaž Rodič. Mobile agents for distributed decision support systems. *Manag. Inf. Syst.*, 6(1) :020–027, 2011. (Cité en page 48)
- [RUFP⁺] Donald Rodriguez-Ubeda, Dora-Luz Flores, Luis Palafox, Manuel Castanon-Puga, Carelia Gaxiola-Pacheco, and Ricardo Rosales. An extended reasoning cycle algorithm for bdi agents. (Cité en page 48)
- [SCL⁺10] I-Fang Su, Yu-Chi Chung, Chiang Lee, et al. Top-k combinatorial skyline queries. In *International Conference on Database Systems for Advanced Applications*, pages 79–93. Springer, 2010. (Cité en page 29), (Cité en page 35), (Cité en page 37)
- [Sim60] H.A. Simon. *The New Science of Management Decision*. Ford distinguished lectures, v. 3. Harper & Row, 1960. (Cité en page 9)
- [SWC⁺02] Jung P Shim, Merrill Warkentin, James F Courtney, Daniel J Power, Ramesh Sharda, and Christer Carlsson. Past, present, and future of decision support technology. *Decision support systems*, 33(2) :111–126, 2002. (Cité en page 47), (Cité en page 48)
- [Tag11] Noria Taghezout. *Conception et développement d’un système multi-agent d’aide à la décision pour la gestion de production dynamique*. PhD thesis, Toulouse 3, 2011. (Cité en page 24)
- [TAZ09a] Noria Taghezout, Abdelkader Adla, and Pascale Zaraté. A multi-agent framework for group decision support system : application to a boiler combustion management system (glz). *International Journal of Software Engineering and Its Applications*, 3(2) :9–20, 2009. (Cité en page 23)
- [TAZ09b] Noria Taghezout, Abdelkader Adla, and Pascale Zaraté. A multi-agent framework for group decision support system : application to a boiler combustion management system (glz). *International Journal of Software Engineering and Its Applications*, 3(2) :9–19, 2009. (Cité en page 24), (Cité en page 46)
- [TEO⁺01] Kian-Lee Tan, Pin-Kwang Eng, Beng Chin Ooi, et al. Efficient progressive skyline computation. In *VLDB*, volume 1, pages 301–310, 2001. (Cité en page 29)
- [Van10] Jan Vanthienen. Rules as data : Decision tables and relational databases. *Business Rules Journal*, 11(1), 2010. (Cité en page 56)
- [VW95] Jan Vanthienen and Geert Wets. Integration of the decision table formalism with a relational database environment. *Information Systems*, 20(7) :595–616, 1995. (Cité en page 56)

- [WYL15] Chen-Shu Wang, Heng-Li Yang, and Shiang-Lin Lin. To make good decision : A group dss for multiple criteria alternative rank and selection. *Mathematical Problems in Engineering*, 2015, 2015. (Cité en page 53), (Cité en page 54)
- [Yin14a] Song Yingdong. Design of distributed support decision-making system model based on multi-agent technique for sporting event. In *2014 Fifth International Conference on Intelligent Systems Design and Engineering Applications*, pages 356–359. IEEE, 2014. (Cité en page 24)
- [Yin14b] Song Yingdong. Design of distributed support decision-making system model based on multi-agent technique for sporting event. In *Fifth ICISDEA*, pages 356–359. IEEE, 2014. (Cité en page 24), (Cité en page 46)
- [YLP⁺19] Wenhui Yu, Jinfei Liu, Jian Pei, Li Xiong, Xu Chen, and Zheng Qin. Efficient contour computation of group-based skyline. *IEEE Transactions on Knowledge and Data Engineering*, 2019. (Cité en page 36), (Cité en page 37)
- [YQL⁺17] Wenhui Yu, Zheng Qin, Jinfei Liu, Li Xiong, Xu Chen, and Huidi Zhang. Fast algorithms for pareto optimal group-based skyline. In *proc. of the 2017 ACM on Conf. on Information and Knowledge Management*, pages 417–426. ACM, 2017. (Cité en page 36), (Cité en page 37)
- [YZM⁺17] Zhibang Yang, Xu Zhou, Jin Mei, Yifu Zeng, Guoqing Xiao, and Guo Pan. Identifying most preferential skyline product combinations. *International Journal of Pattern Recognition and Artificial Intelligence*, 31(11) :1759022, 2017. (Cité en page 29), (Cité en page 35), (Cité en page 37)
- [YZZ⁺16] ZhiBang Yang, Xu Zhou, Yifu Zeng, Fei Zeng, and Yantao Zhou. Identifying most preferential skyline product combinations under price promotion. In *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pages 1824–1828. IEEE, 2016. (Cité en page 29), (Cité en page 35), (Cité en page 37)
- [ZGHW18] Kaiqi Zhang, Hong Gao, Xixian Han, and Jinbao Wang. Finding k-dominant g-skyline groups on high dimensional data. *IEEE Access*, 6 :58521–58531, 2018. (Cité en page 35), (Cité en page 37)
- [ZLH⁺13] Nan Zhang, Chengkai Li, Naeemul Hassan, Sundaresan Rajasekaran, and Gautam Das. On skyline groups. *IEEE Transactions on Knowledge and Data Engineering*, 26(4) :942–956, 2013. (Cité en page 29)
- [ZLH⁺14] Nan Zhang, Chengkai Li, Naeemul Hassan, Sundaresan Rajasekaran, and Gautam Das. On skyline groups. *IEEE Transactions on Knowledge and Data Engineering*, 26 :942–956, 2014. (Cité en page 35), (Cité en page 37), (Cité en page 74)
- [ZLLX19a] H. Zhu, X. Li, Q. Liu, and Z. Xu. Top-k dominating queries on skyline groups. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2019. (Cité en page 35), (Cité en page 37)
- [ZLLX19b] Haoyang Zhu, Xiaoyong Li, Qiang Liu, and Zichen Xu. Top-k dominating queries on skyline groups. *IEEE Transactions on Knowledge and Data Engineering*, 32(7) :1431–1444, 2019. (Cité en page 29)
- [ZLLZ19] H. Zhu, X. Li, Q. Liu, and H. Zhu. Computing skyline groups : an experimental evaluation. *Tsinghua Science and Technology*, 24(2) :171–182, April 2019. (Cité en page 36), (Cité en page 37)

-
- [ZLYL18] Xu Zhou, Kenli Li, Zhibang Yang, and Keqin Li. Finding optimal skyline product combinations under price promotion. *IEEE Transactions on Knowledge and Data Engineering*, 31(1) :138–151, 2018. (Cité en page 36), (Cité en page 37)
- [ZSF⁺05] Xuan F Zha, Ram D Sriram, Marco Gero Fernández, Farrokh Mistree, et al. Knowledge-intensive collaborative decision support for design process, part i : A hybrid decision model and multi-agent framework. 2005. (Cité en page 23)
- [ZZL⁺17] Haoyang Zhu, Peidong Zhu, Xiaoyong Li, Qiang Liu, and Peng Xun. Parallelization of group-based skyline computation for multi-core processors. *Concurrency and Computation : Practice and Experience*, 29(18), 2017. (Cité en page 36), (Cité en page 37)
- [ZZLL17a] Haoyang Zhu, Peidong Zhu, Xiaoyong Li, and Qiang Liu. Computing skyline groups : An experimental evaluation. In *proc. of the ACM Turing 50th Celebration Conf. - China*, ACM TUR-C '17, pages 48 :1–48 :6, New York, NY, USA, 2017. ACM. (Cité en page 36), (Cité en page 37)
- [ZZLL17b] Haoyang Zhu, Peidong Zhu, Xiaoyong Li, and Qiang Liu. Top-k skyline groups queries. In *EDBT*, pages 442–445, 2017. (Cité en page 29), (Cité en page 36)
- [ZZLL17c] Haoyang Zhu, Peidong Zhu, Xiaoyong Li, and Qiang Liu. Top-k skyline groups queries. In *EDBT*, pages 442–445, 2017. (Cité en page 32), (Cité en page 35), (Cité en page 37)

Je présente ici quelques notions indispensables à la lecture du dernier Chapitre. Nous insistons sur l'aspect sémantique (plutôt que l'aspect mathématique formel) ainsi que sur usage. Cependant, pour les lecteurs désireux d'approfondissements, des références utiles et récentes sont données.

.1 Annexe - Logique floue (ou Théorie des ensembles flous)

Le concept d'ensemble flou a été introduit par Zadeh [Zad65]. L'intention de l'auteur est de généraliser la notion classique d'ensemble et la proposition d'accommoder le flou pour représenter des classes ou des ensembles d'objets dont les frontières ne sont pas clairement définies. Ces ensembles permettent de décrire des transitions graduelles entre acceptation totale et rejet absolu. Des exemples typiques de ces classes floues sont ceux décrits à l'aide d'adjectifs ou d'adverbes du langage naturel, telle que "grand individu". Le qualificatif "grand" est un prédicat dont l'extension est un ensemble mal défini qui n'a pas de limite précise. Il n'y a pas de frontière marquée entre l'appartenance indéniable et la non-appartenance absolue à l'ensemble des individus grands, mais une transition graduelle.

Formellement, Un ensemble flou A sur le référentiel X est défini par sa fonction d'appartenance qui associe à chaque élément x dans X un nombre réel dans l'intervalle unité $[0, 1]^3$:

$$\mu_A : X \longrightarrow [0, 1] \quad (7)$$

La valeur de la fonction d'appartenance de l'élément x , i.e., $\mu_A(x)$, représente le degré d'appartenance de x à A . En particulier, $\mu_A(x) = 1$ et $\mu_A(x) = 0$ correspondent respectivement à l'appartenance totale et la non-appartenance absolue. Lorsque $0 < \mu_A(x) < 1$, on parle d'appartenance nuancée ou partielle. Par exemple, la classe des individus "*grands*" peut être représentée par un ensemble flou dont la fonction d'appartenance est donnée par la Figure 9.

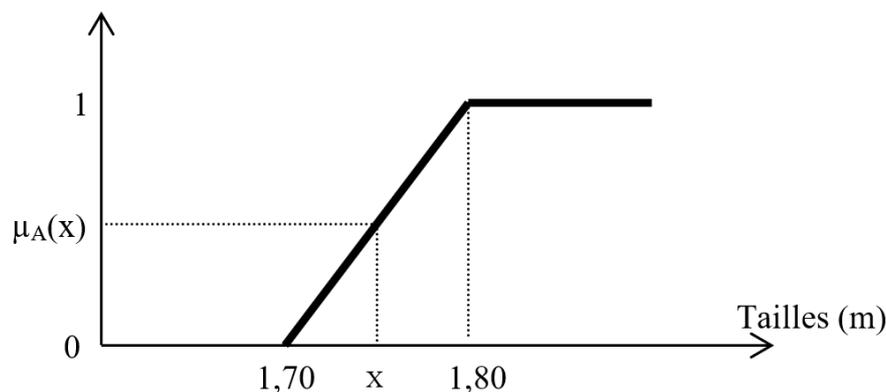


FIGURE 9 – Représentation du prédicat "grand" dans le référentiel des tailles des personnes

Il apparaît qu'une taille en dessous de 1,70 m ne correspond pas du tout au prédicat grand, qu'une taille supérieure ou égale à 1,80 m est en complète adéquation avec l'idée de grand, et que la taille 1,78m est considérée plus conforme à l'idée de *grand* que la taille 1,75 m. On peut observer que l'extension du prédicat graduel grand est munie d'une relation de préférence entre tailles plus ou moins adéquates ou typiques du prédicat.

3. L'intervalle $[0, 1]$ peut être remplacé par une échelle totalement ordonnée, bornée supérieurement et inférieurement. On emploie le plus souvent en pratique l'intervalle réel $[0, 1]$, mais cette échelle peut aussi ne comporter qu'un nombre fini de niveaux.

.1.1 Formes de la gradualité

La gradualité d'un prédicat (qui signifie que sa satisfaction n'est pas tout ou rien mais une question de degré), semble être liée à deux faits :

- À l'existence de référentiels continus : Par exemple, un ensemble de tailles pour le prédicat grand. Pour être conforme au sens commun et à la perception humaine, il ne peut exister un seuil précis s^* du référentiel tel que en dessous de s^* le prédicat est totalement satisfait et, au dessus de s^* le prédicat ne s'applique pas du tout. La transition entre appartenance totale et non appartenance, n'est pas brutale mais au contraire graduelle.

- À l'idée de la typicalité : Il peut exister des classes d'objets où certains objets de ces classes sont plus typiques que d'autres. Par exemple, dans la classe oiseau, les moineaux sont des oiseaux plus typiques que les pingouins. Ainsi, la classe oiseau n'est pas juste un ensemble, elle est un ensemble partiellement ordonné par la relation "oiseau plus typique que".

.1.2 Sémantique du degré d'appartenance

La fonction d'appartenance d'un ensemble flou A permet un codage simple de trois notions de base bien différentes qu'il convient de ne pas confondre : la similarité, la préférence et l'incertitude. Ainsi $\mu_A(x)$ peut exprimer :

- Un degré de similarité : d'un objet x avec les prototypes d'une classe. Dans ce cas, $\mu_A(x)$ est le degré de proximité de x des éléments prototypiques du concept décrit par A. Historiquement, c'est la plus ancienne sémantique des degrés d'appartenance.

- Un degré de préférence : d'un objet x au sens de plusieurs critères et de contraintes flexibles. Dans ce cas, A représente une contrainte (critère) flexible restreignant un ensemble d'objets plus ou moins préférés et $\mu_A(x)$ évalue l'intensité de préférence en faveur d'un objet x.

- Un degré d'incertitude : qu'un objet x dont on ne connaît qu'une description imprécise soit bien celui qu'on cherche à identifier. Dans ce cas, A traduit une description d'états de connaissances incomplets. $\mu_A(x)$ est le degré de possibilité qu'un paramètre w prenne la valeur x, étant donné l'état de connaissance incomplet "w est A".

.1.3 Concepts de base

L'ensemble des éléments appartenant tout-à-fait à A, c'est-à-dire $x \text{ in } X / \mu_A(x) = 1$, est appelé le noyau de A, noté $C(A)$; l'ensemble des éléments qui appartiennent au moins quelque peu à A, c'est-à-dire $\{x \in X / \mu_A(x) > 0\}$, est appelé le support de A, noté $S(A)$. Pour $\alpha \in]0, 1]$, l'ensemble classique $A = \{x \in X / \mu_A(x) \geq \alpha\}$ est appelé une coupe de niveau α (ou α -coupe tout court) de A. Soit maintenant $\alpha \in [0, 1[$, l'ensemble classique $A_\alpha = \{x \in X / \mu_A(x) > \alpha\}$ est appelé une α -coupe forte (ou stricte) de A. On a immédiatement $A_0 = S(A)$ et $A_1 = C(A)$. En pratique, les fonctions d'appartenance sont souvent de forme trapézoïdale représentées par le quadruplet (A_1, A_2, a_1, a_2) où $[A_1, A_2]$ (resp. $[A_1 - a_1, A_2 + a_2]$) représente le noyau (resp. le support). Par exemple, dans la Figure 4.1, la sémantique de l'ensemble flou grand est donnée par $(1.80, \infty, 0.10, 0)$. Si X est un ensemble fini $\{x_1, \dots, x_n\}$, et A un ensemble flou de X alors on utilise souvent la notation

$$A = \{\mu_1/x_1 + \dots + /x_n\}, \tag{8}$$

où le terme μ_i/x_i , $i=1,n$, signifie que μ_i est le degré d'appartenance de x_i à A, et le signe '+' représente l'union.

Les opérations et relations ensemblistes usuelles sont étendues aux ensembles flous, par exemple (A et B sont deux ensembles flous définis sur le même référentiel X) :

- **Inclusion** : $A \subseteq B \iff \forall x \in X, \mu_A(x) \leq \mu_B(x)$.

- **Égalité** : $A = B \iff \forall x \in X, \mu_A(x) = \mu_B(x)$.

- **Complémentation** : Le complémentaire d'un ensemble flou A, noté \bar{A} , est défini par : $\forall x \in X, \mu_{\bar{A}}(x) = 1 - \mu_A(x)$

- **Intersection** : L'ensemble flou $A \cap B$ est défini par $\mu_{A \cap B}(x) = \square(\mu_A(x), \mu_B(x))$ où \square est une norme triangulaire (opérateur de $[0, 1] \times [0, 1] \rightarrow [0, 1]$) comme le minimum et le produit.

- **Union** : L'ensemble flou $A \cup B$ est défini par $\mu_{A \cup B}(x) = L(\mu_A(x), \mu_B(x))$ où L est une co-norme triangulaire (opérateur de $[0, 1] \times [0, 1] \rightarrow [0, 1]$) comme le maximum et la somme probabiliste (i.e., $L(x, y) = x + y - x \cdot y$).

L'implication floue, notée \implies_f , est une fonction définie de $[0, 1] \times [0, 1] \rightarrow [0, 1]$. Deux familles d'implications sont généralement utilisées :

- **Les S-implications** : qui constituent une généralisation de l'implication matérielle $x \implies y = \neg x \vee y$. Elle est définie par $x \implies_f y = \square(1 - x, y)$. Comme exemples de cette famille, on peut citer l'implication de Kleene-Dienes ($x \implies_f y = \max(1 - x, y)$) et celle de Lukasiewicz ($x \implies_f y = \min(1, 1 - x + y)$).

- **Les R-implications** : qui vérifient la relation suivante $x \implies_f y = \sup_{[0,1]} u \mid \square(x, u) \leq y$. Les plus connues sont l'implication de Gödel ($x \implies_f y = 1$ si $x \leq y$, y/x sinon) et celle de Goguen ($x \implies_f y = 1$ si $x \leq y$, y/x sinon).

Une quantité floue est un ensemble flou normalisé Q (i.e., $\exists x_0 \in X, \mu_Q(x_0) = 1$) de la droite réelle. Elle permet de modéliser une quantité vaguement perçue ou imprécisément connue, vue comme un ensemble de valeurs possibles aux contours mal délimités. Les intervalles flous (resp. nombres flous) sont des quantités floues convexes. Les opérations arithmétiques étendues aux intervalles flous, comme l'addition (\oplus) et le produit (\otimes), ont aussi été définies. Si $M = (m_1, m_2, a, b)$ et $N = (n_1, n_2, c, d)$ sont deux intervalles flous alors $M \oplus N = (m_1+n_1, m_2+n_2, a+c, b+d)$ (resp. $M \otimes N = (m_1n_1, m_2n_2, m_1c + n_1a, m_2d + n_2b)$).

Résumé

Les systèmes d'aide à la prise de décision (ou DSS pour Decision Support Systems) collectent et compilent des informations à partir d'un large éventail de sources afin de faciliter la résolution de problèmes et la prise de décisions. Dans de nombreux domaines d'application (choix d'investissements, domaine médical, domaine militaire, etc.), les sources d'informations ne sont plus localisées sur un même site mais réparties sur de multiples sites pouvant communiquer entre eux pour contribuer à la prise de décisions. Pour faire face à ce défi, des systèmes d'aide à la décision distribués (ou DDSS pour Distributed Decision Support Systems) ont émergé. Par ailleurs, dans certaines applications, la décision est de nature collective, c'est-à-dire, elle ne concerne pas un seul utilisateur/choix/objet mais un groupe d'utilisateurs/choix/objets.

Le travail de cette thèse s'inscrit dans le cadre de décisions de groupe dans un environnement distribué. L'approche que nous avons développée s'appuie sur la technologie agents et sur le paradigme Skyline. Les agents permettent de représenter la diversité des acteurs concernés par la décision ainsi que leurs interactions, d'une part, et de proposer des solutions robustes tout en s'adaptant à des environnements qui peuvent être imprévisibles et dynamiques, d'autre part. L'approche Skyline constitue un outil approprié pouvant aider l'utilisateur à prendre des décisions intelligentes en présence de données multidimensionnelles et lorsque différents, et souvent contradictoires, critères sont à prendre en compte.

Dans une première contribution de la thèse, nous avons proposé une approche multi-agents pour la modélisation des DDSS. La spécificité de notre proposition est sa généralité, son dynamisme et son adaptabilité aux changements rapides de l'environnement avec l'agent d'observation. Le système proposé est capable de prédire ou de détecter des pannes dans les plus brefs délais et mettre ainsi en œuvre des actions de récupération à l'aide d'agents miroirs. Il est à noter que l'approche intègre aussi une phase d'apprentissage pour l'identification des problèmes de décision similaires. Une validation formelle de l'approche a aussi été réalisée. Dans un second temps, nous nous sommes intéressés à l'étude d'un modèle de Skyline de groupe centré utilisateur et offrant des procédures de contrôle des résultats afin de garantir des décisions meilleures et pertinentes. Cette contribution a porté sur la sémantique du modèle et aussi sur l'aspect algorithmique pour le calcul du Skyline de groupe relaxé. Enfin, une série d'expérimentations ont été menées, sur des données synthétiques et réelles, pour valider et comparer notre proposition.

Mots-clés : DDSS, Décision collective, Multi-agents, Skyline de groupe, Relaxation, Préférences utilisateurs. .

Abstract

Decision Support Systems (DSS) collect and compile information from a wide variety of sources to help in problem solving and decision making. In many fields of application (choice of investments, medical field, military field, etc.), the sources of information are no longer located on the same place but distributed over multiple sites that can communicate with each other to contribute to the decision. To address this challenge, Distributed Decision Support Systems (DDSS) have emerged. Moreover, in some applications, the decision is of a collective nature, that is to say, it does not concern a single user/choice/object but a group of users/choices/objects.

The work of this thesis is part of group decisions in a distributed environment. The approach we have developed is based on agent technology and the Skyline paradigm. The agents make it possible to represent the diversity of the actors concerned by the decision as well as their interactions, on the one hand, and to propose robust solutions while adapting to environments which can be unpredictable and dynamic, on the other hand. The Skyline approach is an appropriate tool that can help the user to make intelligent decisions in the presence of multidimensional data and when different, and often contradictory, criteria are to be taken into account.

In a first contribution of the thesis, we proposed a multi-agent approach for modeling DDSS. The specificity of our proposal is its generality, its dynamism and its adaptability to rapid changes in the environment with the observation agent. The proposed system is able to predict or detect failures in the shortest possible time and thus implement recovery actions using mirror agents. It should be noted that the approach also includes a learning phase for the identification of similar decision problems. A formal validation of the approach was also carried out. In a second step, we were interested in the study of a user-centered group Skyline model and offering procedures for controlling the results in order to guarantee better and relevant decisions. This contribution focused on the semantics of the model and also on the algorithmic aspect for the calculation of the relaxed group Skyline. Finally, a series of experiments were carried out, on synthetic and real data, to validate and compare our proposal.

Keywords : DDSS, Collective decision, Multi-agents, Group skyline, Relaxation, User preferences. .

Secteur de recherche : Informatique et applications

Secteur de recherche : IA