

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

UNE APPROCHE D'OPTIMISATION PAR ESSAIM DE PARTICULES POUR LA  
RECHERCHE EN MÉMOIRE DE CAS

THÈSE  
PRÉSENTÉE  
COMME EXIGENCE PARTIELLE  
DU DOCTORAT EN INFORMATIQUE COGNITIVE

PAR  
NABILA NOUAOURIA

NOVEMBRE 2013

UNIVERSITÉ DU QUÉBEC À MONTRÉAL  
Service des bibliothèques

Avertissement

La diffusion de cette thèse se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»



## REMERCIEMENTS

Merci à mes directeurs de thèse Pr. Mounir Boukadoum et Pr. Robert Proulx pour avoir supervisé ce travail.

Merci à tous ceux qui m'ont soutenue durant mes études au DIC.

Merci aussi à ma famille pour son support continu.

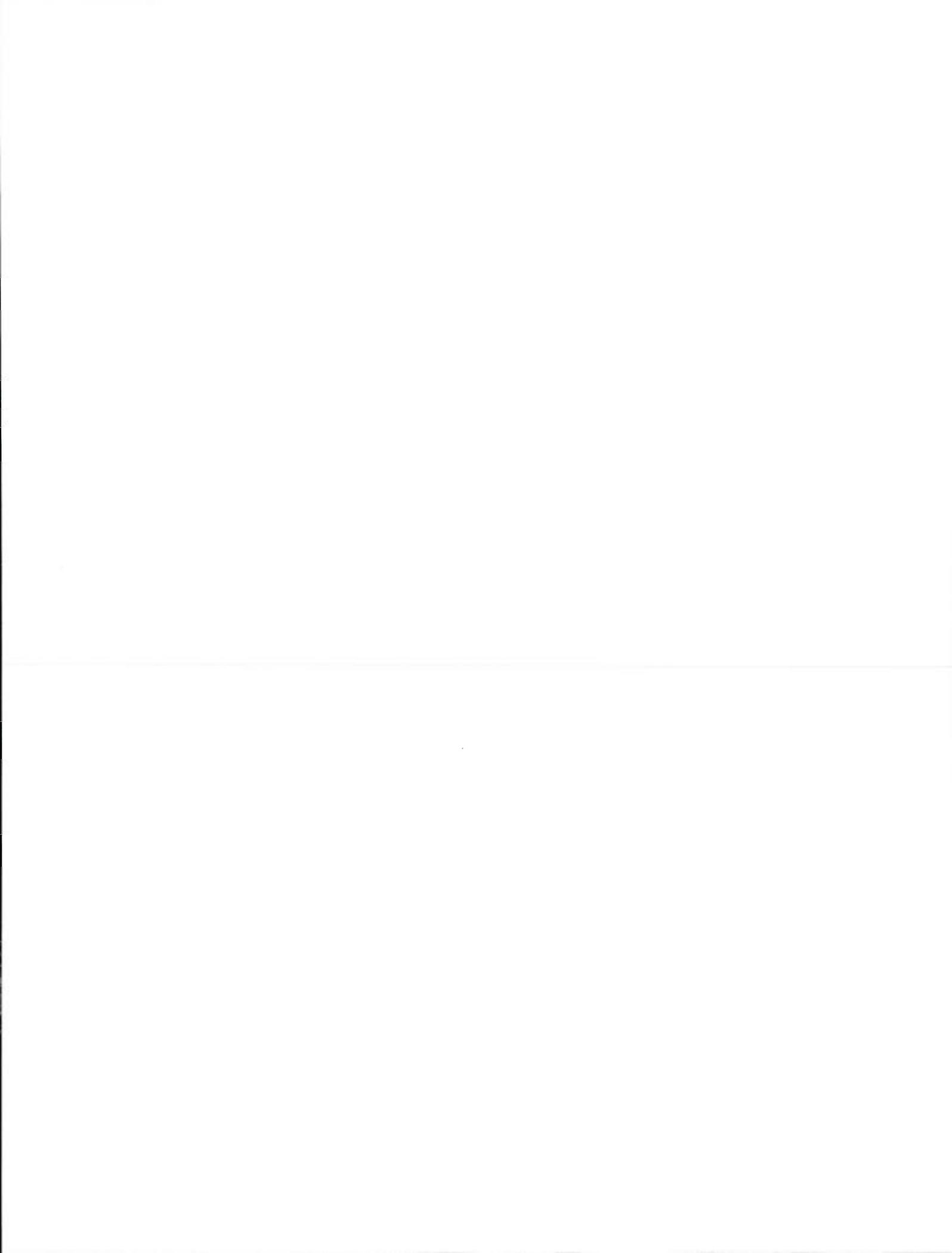


## TABLE DES MATIÈRES

TABLE DES MATIÈRES .....	III
LISTE DES FIGURES.....	VII
LISTE DES TABLEAUX.....	XI
RÉSUMÉ .....	XIII
INTRODUCTION .....	1
0.1 Motivations et aperçu général.....	1
0.2 Problématique et thèse.....	2
0.3 Contributions .....	4
0.4 Plan du document .....	6
CHAPITRE I	
1.1 Introduction.....	11
1.2 Présentation générale.....	11
1.3 Omniprésence du CBR .....	13
1.4 Modélisation des connaissances en CBR .....	14
1.4.1 Le Cas.....	16
1.4.2 La Base de Cas .....	22
1.4.3 Modélisation des Tâches .....	23
1.5 Méthodes de construction des systèmes CBR .....	27
1.5.1 Conception de la Base de cas .....	27
1.5.2 Développement initial de la Base de cas .....	28
1.5.3 Développement continu et maintenance .....	28
1.6 Situation du paradigme CBR.....	28
1.6.1 CBR et Analogie .....	29
1.6.2 CBR et Apprentissage automatique .....	30
1.6.3 CBR et Réutilisation.....	31

1.7	La facette cognitive du Raisonnement basé cas .....	33
1.8	La génération de plan en CBR .....	36
1.8.1	Modèle de connaissances et cycle d'inférence en CBP.....	39
1.8.2	Cadre spécifique à la thèse et caractérisation cognitive du CBP.....	45
1.9	Conclusion .....	51
CHAPITRE II		
2.1	Introduction.....	55
2.2	La genèse de la Swarm Intelligence.....	55
2.3	L'inspiration éthologique.....	57
2.4	Modèle de l'essaim .....	61
2.5	La PSO de base .....	69
2.6	Variantes et hybridations de la PSO.....	73
2.7	Situation de la PSO .....	76
2.8	Conclusion .....	77
CHAPITRE III		
3.1	Introduction.....	81
3.2	La mémoire de cas dans la littérature.....	81
3.2.4	À la recherche du bon cas .....	87
3.4	Architecture de mémoire proposée .....	92
3.5	Modélisation.....	95
3.5.1	Mise en équation.....	95
3.5.2	La PSO à valeurs mixtes.....	98
3.6	Conclusion .....	106
CHAPITRE IV		
4.1	Introduction.....	109
4.2	Ancrage méthodologique et justification du choix .....	109
4.3.0	Cadre applicatif.....	111
4.3.1	Modélisation .....	113
4.3.2	Diagramme de classes.....	119
4.3.3	Diagrammes de séquences.....	120
4.3.4	Diagramme état/transition. ....	121

4.4	Architecture générale.....	122
4.5	Modélisation des connaissances pour le CBP .....	124
4.6	Conclusion .....	129
<b>CHAPITRE V</b>		
5.1	Introduction.....	133
5.2	Démarche.....	133
5.3	Connaissances utilisées.....	135
5.3.1	Dans le cadre des étapes 1 et 2 : « Développement d'un algorithme PSO pour la prise en charge de données mixtes » et « Premier modèle de mémoire basé PSO ».....	135
5.3.2	Pour l'étape 3 : « un modèle de mémoire basé PSO intégrant le critère d'adaptabilité ».....	141
5.3.3	Étape 4 : « une modélisation CBR pour la planification ».....	147
5.4	Environnement de programmation .....	151
5.5	Conclusion .....	152
<b>CHAPITRE VI</b>		
6.1	Introduction.....	155
6.2	Validation des mécanismes d'interprétation.....	155
6.3	Résultats de la PSO pour la recherche en mémoire.....	156
6.3.1	Résultats de la recherche basée similarité .....	156
6.3.2	Résultats de la PSO et la RGR .....	159
6.3.3	Résultats de la PSO pour le CBP .....	161
6.4	Discussion.....	163
6.5	Conclusion .....	168
<b>CHAPITRE VII</b>		
<b>CONCLUSION GÉNÉRALE</b>		
7.1	Rappel des objectifs.....	171
7.2	Évaluation et critiques .....	172
7.3	Retour sur les contributions.....	175
7.4	Perspectives .....	177
APPENDICE A Classification basée PSO pour les données mixtes .....		179
Bibliographie .....		189



## LISTE DES FIGURES

Figure	Page
1.1 Cycle CBR proposé dans (Aamodt et Plaza, 1994)	15
1.2 Un cas C est composé du problème Pb, du raisonnement R et de la solution S.	20
1.3 La décomposition (incomplète) de la tâche Raisonner à partir de cas. 24	
1.4 Techniques de raisonnement pour l'apprentissage.	31
1.5 Les axes génie logiciel/intelligence artificielle	32
1.6 Cycle du CBP	41
1.7 La tâche de remémoration de plan.	42
1.8 La tâche de réutilisation de plan.	43
1.9 La tâche de révision de plan	44
1.10 La tâche de stockage de plan.	45
1.11 Principaux domaines d'études des descriptions d'itinéraires	46
2.1 Modèles de comportements collectifs: (a)essaim (b) torus (c)groupe parallèle dynamique et (d) groupe hautement parallèle	58
2.2 Le simple schéma d'action de l'essaim.	59
2.3 Principes du comportement collectif de l'essaim.	61
2.4 Organigramme de la PSO standard	71
3.1 La base de cas vue comme un réseau de cas.	83
3.2 Principe d'une mémoire associative	85

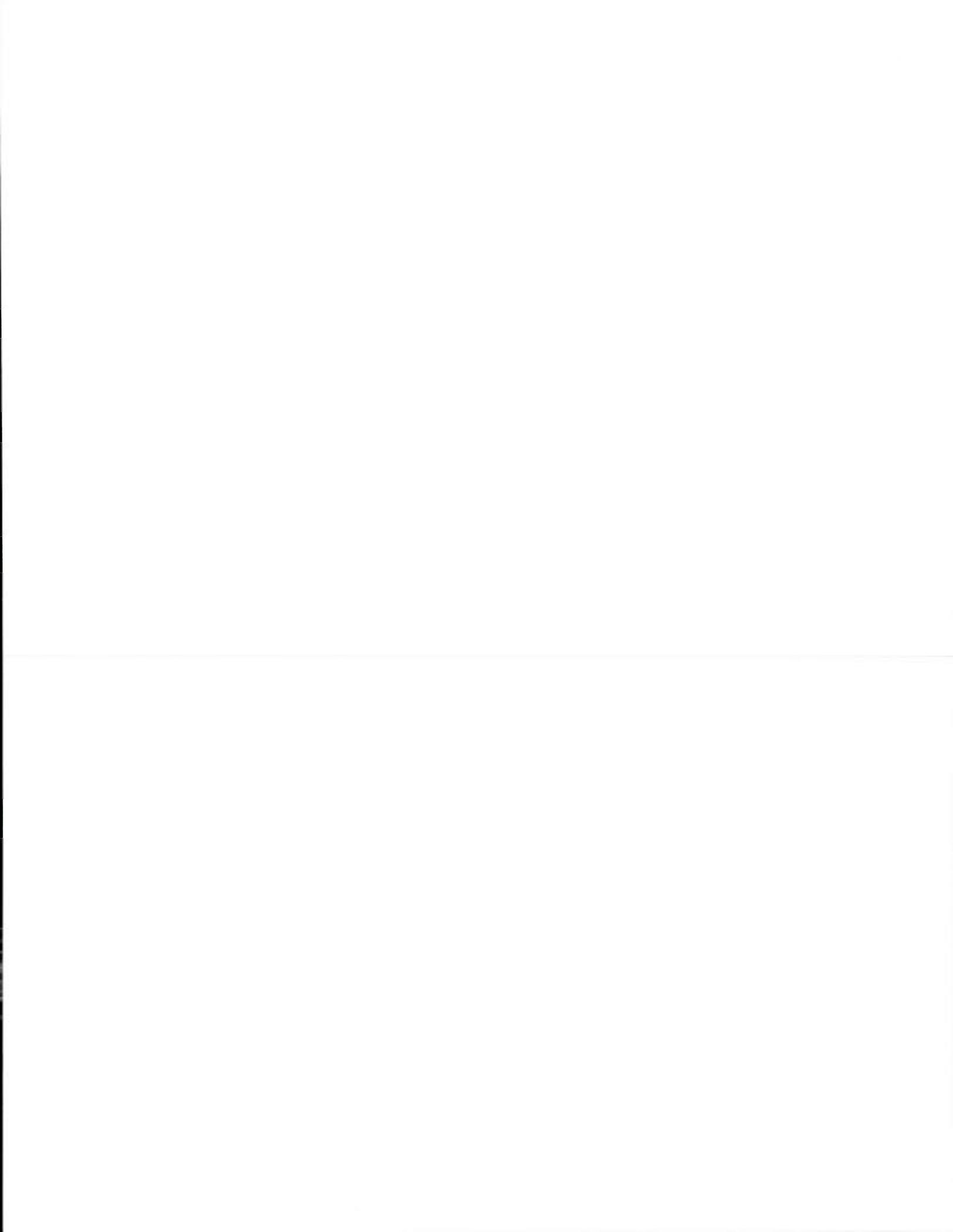
3.3	Espace problème/Espace solution .....	88
3.4	Compromis entre recherche et réutilisation .....	89
3.5	Décomposition de la tâche 'Réutiliser' .....	90
3.6	Architecture de mémoire proposée.....	94
3.7	Mécanismes d'interprétation entre l'espace de recherche et l'espace de description. ....	102
3.8	Organigramme de la PSO à valeurs mixtes.....	104
4.1	Cas d'utilisation global .....	115
4.2	Les cas d'utilisation organisés en package .....	116
4.3	Diagramme de classes. ....	120
4.4	Diagramme de séquence1.....	121
4.5	Diagramme de séquence2.....	121
4.6	Diagramme d'état/transition. ....	122
4.7	. Architecture générale. ....	123
4.8	Un cas C est composé du problème Pb, du raisonnement R et de la solution S.....	125
4.9	Un cas C pour la Planification.....	125
6.1	Étude comparative de la similarité (axe des y) entre recherché linéaire et recherche basée PSO pour 1 <sup>le</sup> groupe de requêtes (axe des x). ....	157
6.2	Étude comparative du temps de recherche (en secondes sur l'axe des y) de la recherche linéaire et de la recherche basée PSO pour le 1 <sup>er</sup> groupe de requêtes (axe des x). ....	158
6.3	Étude comparative de la similarité (axe des y) de la recherche linéaire et de la recherche PSO pour le 2 <sup>nd</sup> groupe de requêtes (axe des x). ....	159
6.4	Similarité par recherche linéaire et recherche guidée réutilisabilité basée PSO pour des requêtes non existantes dans la base de cas (axe des x). ....	161

6.5	Recherche linéaire guidée similarité et recherche guidée réutilisabilité basée PSO pour des requêtes non existantes dans la base de cas (axe des x). .....	163
A.1	Résultats pour la base 'Adult': La précision et le temps d'apprentissage pour la PSO avec dispersion et confinement avec un essaim de 10 particules and $T_{\max}$ variant de 10 à 200 itérations. ....	182
A.2	Résultats pour la base 'Adult': Précision et temps d'apprentissage pour la PSO avec mécanismes de dispersion et de confinement pour un essaim de 40 particules et 10 rotations (axe des x) en validation croisée. ....	183
A.3	Résultats pour la base CMC: Précision et temps d'apprentissage pour la PSO avec mécanismes de dispersion et de confinement pour un essaim de 10 particules et différents nombres d'itérations (axe des x) en validation croisée (chaque valeur sur la courbe est la moyenne de 10 rotations, chacune d'elles étant la moyenne de 10 essais). ....	184
A.4	Résultats pour la base CMC: Précision et temps d'apprentissage pour la PSO avec mécanismes de dispersion et de confinement pour 1000 itérations et différentes tailles de l'essaim (axe des x) en validation croisée (chaque valeur sur la courbe est la moyenne de 10 rotations, chacune d'elles étant la moyenne de 10 essais). ....	185
A.5	Résultats pour la base 'Abalone': Précision et temps d'apprentissage pour la PSO avec mécanismes de dispersion et de confinement pour un essaim de 10 particules et différents nombres d'itérations (axe des x) en validation croisée (chaque valeur sur la courbe est la moyenne de 10 rotations, chacune d'elles étant la moyenne de 10 essais). ....	186



## LISTE DES TABLEAUX

Tableau .....	Page
1.1 Quelques domaines de recherche liés à la résolution paresseuse de problème. ....	33
3.1 Aperçu des méthodes de recherche en mémoire de cas.....	86
3.2 Tableau partiel des Fréquences pour deux descripteurs .....	104
4.1 Description du cas.....	126
4.2 Description d'un segment.....	126
4.3 Description d'une préférence.....	127
4.4 Aperçu des valeurs possible des EIs.....	127
5.1 Exemple d'heuristique de similarité pour l'attribut catégoriel 'sex' pour la base 'Abalone' .....	140
5.2 Exemple d'heuristique de similarité pour l'attribut catégoriels 'race' pour la base 'Adult'. Le symbole '?' est utilisé pour les données manquantes. ....	140
5.3 Exemples de cas dans GUIDIET .....	143
5.4 Dépendances Partielles Problème/solution dans GUIDIET .....	146
5.5 Numérotation des intersections pour la base PLAN.....	148
5.6 Exemples de cas dans PLAN .....	149
5.7 Exemple d'heuristique de similarité pour les dimensions de type 'intersection'. ....	150
6.1 Requêtes utilisée pour la base GUIDIET.....	160
6.2 Requêtes pour la base PLAN.....	162



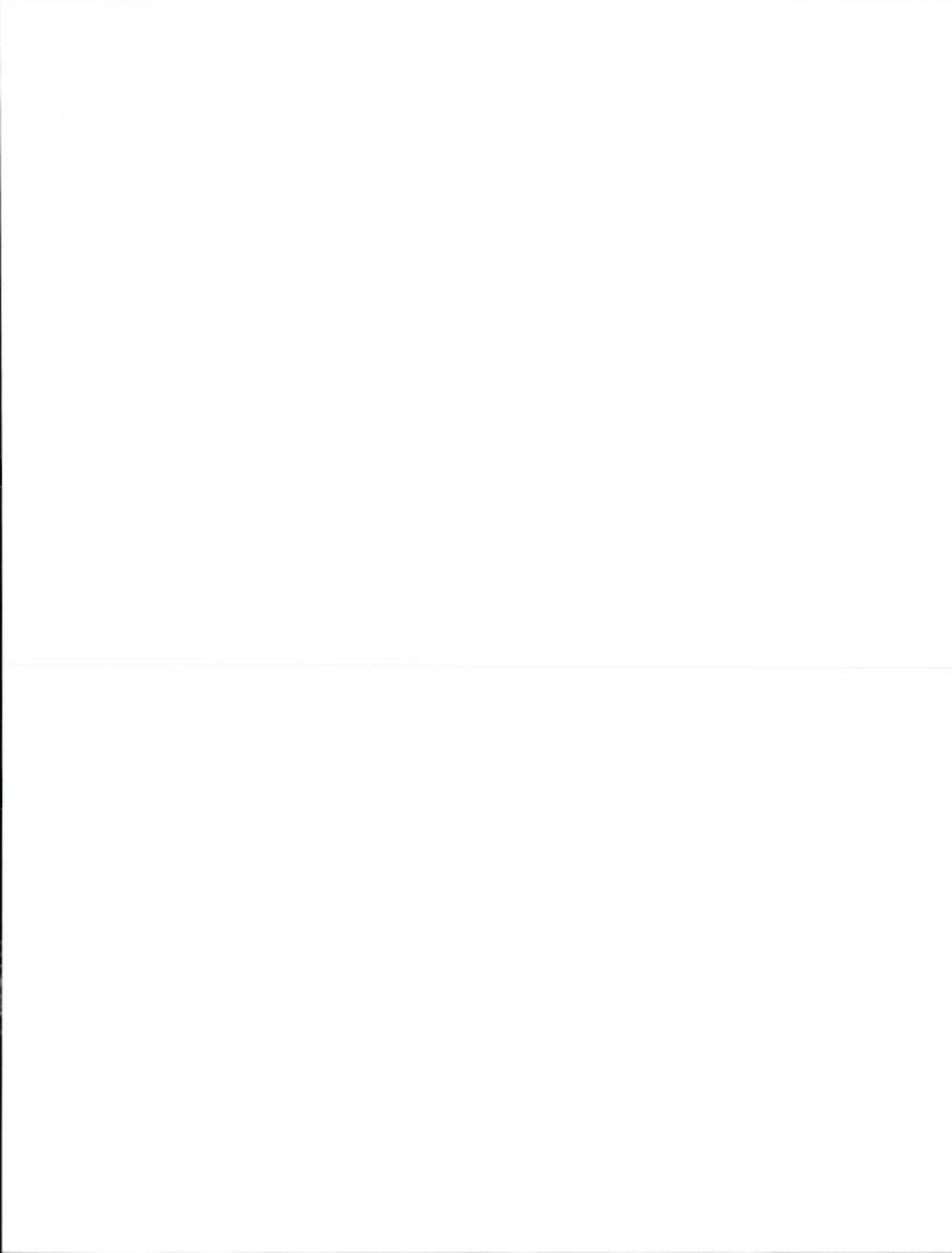
## RÉSUMÉ

Dans le présent travail, nous nous intéressons à la modélisation du mécanisme cognitif de remémoration que nous plaçons au cœur d'un processus de résolution de problème par réutilisation de cas résolus et mémorisés. Le but est d'exploiter de nouvelles pistes de recherche et d'exploration de mémoire applicables aux problèmes de grande dimension et ceux pour lesquels un modèle formel précis n'existe pas ou est difficile à établir. Nous proposons la modélisation du processus de rappel mnémorique à l'aide d'une structure associative indépendante de l'organisation préalable de l'espace de stockage. En nous inspirant de modèles éthologiques, nous proposons l'utilisation d'un essaim de particules (Particle Swarm Optimization PSO) pour l'exploration d'une mémoire d'anciens cas en vue de les réutiliser pour résoudre un nouveau.

Notre approche de résolution de problème s'inscrit parmi les méthodes dites de second principe où l'emphase est mise sur la réutilisation de solutions ayant déjà servi à des problèmes similaires plutôt que la génération de solution par un processus analytique (résolution de premier principe). Dans le problème qui nous intéresse, la clé du succès est la remémoration du bon cas — celui qui rendra la réutilisation efficace —, donc le plus similaire dans sa partie problème et le plus adaptable dans sa partie solution. Notre approche évite le biais structurel que peut introduire une pré-organisation particulière de la mémoire et, de ce fait, améliore la perception de l'espace de recherche et la généralité du processus de rappel. De plus, le parcours stochastique de l'espace de recherche permet la génération d'un voisinage (sous-ensemble de la base de cas sélectionné selon un critère spécifique) caractérisé par une réutilisabilité forte, puis l'accès direct à ce voisinage dans une mémoire associative de cas.

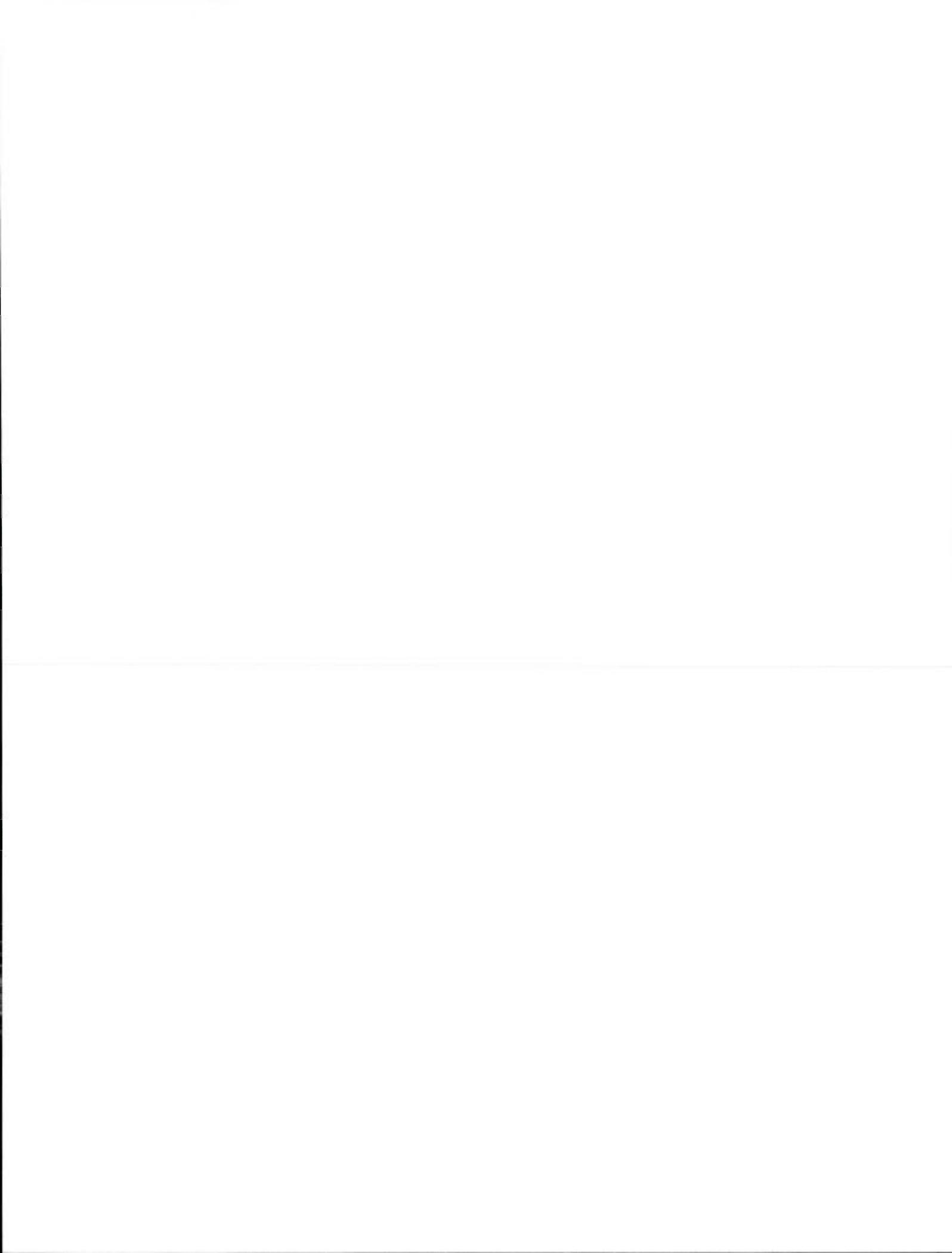
Notre démarche est réalisée à travers un certain nombre de contributions originales dont la mise en œuvre d'un modèle inédit de mémoire basé PSO, l'intégration du critère de réutilisabilité dans le modèle de rappel de mémoire basé PSO, le développement d'un algorithme PSO pour la prise en charge de données mixtes et la mise en œuvre d'un modèle générique pour les connaissances de génération de plan.

Mots clés : Raisonnement à base de Cas (CBR), Optimisation par essaim de particules, Rappel en mémoire de cas, Traitement de données mixtes, Génération de plan basée remémoration.



## INTRODUCTION

1. Motivations et aperçu général
2. Problématique et thèse
3. Contributions
4. Plan de la thèse



**L**a mémoire est au cœur de la cognition humaine, qu'il s'agisse de résolution de problèmes ou d'apprentissage. Elle nourrit notre pensée et se nourrit d'elle. On arrive à résoudre par le souvenir des problèmes déjà rencontrés dans le passé, et les problèmes nouvellement résolus s'ajoutent à notre souvenir. La mémoire est réellement un thème central auquel de nombreux efforts ont été consentis dans diverses disciplines. Ce travail s'inscrit dans cette lignée en exploitant de nouvelles pistes de recherche et d'exploration de la mémoire dans le cadre du raisonnement basé cas (CBR), un paradigme de résolution de problèmes par réutilisation de cas déjà résolus.

### 0.1 Motivations et aperçu général

Le travail présenté rentre dans un cadre méthodologique de second principe, dans notre cas la solution d'un nouveau problème en réutilisant une solution convenant à des situations similaires, soit l'approche CBR. Pour fonctionner correctement, le CBR réutilise des cas stockés dans une structure de mémoire qui représente les problèmes déjà rencontrés dans le domaine. Cet ensemble est filtré lors d'une étape de présélection afin de générer un sous-ensemble ou voisinage de candidats qui répondent à un critère de recherche donné (généralement la similarité). Plus riche est cette mémoire, plus efficace est le système en termes de capacité de résolution. Cependant, la richesse obtenue peut induire un coût de recherche prohibitif pour trouver la bonne expérience à réutiliser. Nous proposons comme axe principal de cette thèse d'adresser ce problème en modélisant le processus de rappel de cas à l'aide d'une structure associative indépendante de toute organisation préalable de l'espace de stockage. Nous nous inspirons d'une analogie éthologique pour développer une méthodologie d'exploration de la mémoire en vue de réutiliser les cas stockés. Pour le type de problème qui nous intéresse et afin de trouver le meilleur cas à réutiliser dans les meilleurs délais, nous ramenons le processus de recherche à un problème d'optimisation pour lequel nous mettons en œuvre un algorithme

d'optimisation par essaim de particules ou PSO (Particle Swarm Optimization) pour sa solution.

D'un point de vue cognitif, nous visons deux objectifs par cette approche : 1) une exploration rapide de la mémoire de cas à travers un dispositif associatif, ce qui continue à nous situer parmi les processus rapides<sup>1</sup> (mais biaisés) de résolution de problème ; 2) Une exploration stochastique de la mémoire de cas basée sur un « mind-ware »<sup>2</sup> probabiliste qui élimine le biais du processus rapide tout en gardant les avantages.

## 0.2 Problématique et thèse

L'atteinte de nos objectifs passe par plusieurs phases, avec une succession de sous-problèmes emboîtés à résoudre. Il faudra en particulier traduire notre approche de résolution de second principe en un problème d'exploration stochastique de l'espace de recherche. Notre thèse s'articule autour de quatre énoncés liés aux sous-problèmes identifiés.

Lors de la première phase d'analyse, la question qui s'impose porte sur la représentation des connaissances qui convient pour un rappel efficace, de même que leur granularité. Nous posons donc comme premier énoncé de thèse : *La clé du succès pour la résolution de problème basée CBR est la remémoration du bon cas.*

Comme deuxième questionnement vient le choix de l'architecture de mémoire à mettre en place pour obtenir un mécanisme de rappel efficace non seulement en termes de temps, mais aussi de précision afin de garantir la qualité de résolution du problème. Il s'agit donc de mettre en place un mécanisme capable, dans les meilleurs délais, de retrouver le bon cas à réutiliser, d'où notre deuxième énoncé de thèse : *Un bon cas est celui qui rend la réutilisation efficace, donc le plus similaire dans sa*

---

<sup>1</sup> Nous faisons ici référence aux processus rapides par opposition aux processus lents de prise de

<sup>2</sup> Terme proposé par David Perkins faisant référence aux règles, données, procédures, stratégies et autres outils cognitifs (connaissance de probabilité, inférences logique et scientifique) qui doivent être retrouvés en mémoire afin de permettre une réflexion rationnelle selon Stanovich (2009).

*partie problème et le plus adaptable dans sa partie solution. Comment atteindre cette efficacité de temps et de précision amendée avec la notion de réutilisabilité?*

La question précédente mène à un troisième niveau d'analyse en vue de concrétiser l'efficacité recherchée du processus de rappel en mémoire en termes de stratégie d'exploration de l'espace de recherche. Bien entendu, l'exhaustivité d'un parcours séquentiel place ce dernier au premier rang de la précision, mais sans le moindre compromis en temps. Quant aux approches s'appuyant sur une structuration préalable de l'espace, elles gagnent en temps, mais perdent en précision (Lenz et al. 1998); sans oublier le biais structurel potentiel qu'elles introduisent par l'imposition d'une organisation préalable des données et d'un schéma de parcours de la mémoire. En effet, l'organisation de cette dernière selon un critère particulier impose une vision qui peut ne pas refléter la réalité dans son intégralité et qui fige la perception à un point de vue imposé par le ou les critères choisis. Dans ce contexte, tout changement de critère mène à la réorganisation de la mémoire (Schaaf, 1996). Notre énoncé de thèse à ce niveau est que : *L'élimination du biais structurel améliore la perception de l'espace de recherche et la qualité du rappel par là même.*

L'énoncé précédent pose la question de parcourir l'espace de recherche selon un critère indépendant de la structure de l'espace. On ramène ainsi le critère de recherche à son vrai rôle qui est de sélectionner ou filtrer, et non d'organiser. L'exploration stochastique ayant porté ses fruits dans divers domaines de l'intelligence artificielle, nous l'appliquons à la recherche de cas. L'idée est d'effectuer un parcours stochastique de l'espace de recherche par lequel la cible à atteindre est identifiée par ses caractéristiques de similarité et d'adaptabilité (nos critères choisis), sans s'appuyer pour cela sur une structure de parcours quelconque. À cet égard, l'énoncé de thèse est : *le parcours stochastique de l'espace de recherche permet la génération du voisinage caractérisé par une réutilisabilité forte, puis l'accès direct à ce voisinage dans une mémoire associative de cas.*

Les quatre énoncés précédents constituent l'épine dorsale de ce travail. Les contributions spécifiques suivent.

### 0.3 Contributions

Les contributions visent à résoudre les problèmes dégagés dans la section précédente. Comme l'objectif de la thèse présente une certaine complexité sur le plan méthodologique, en particulier à cause de deux dimensions PSO et CBR, nous avons adopté une démarche de résolution incrémentale. La charge de résolution a été divisée en étapes visant chacune à résoudre un objectif particulier relié à l'une ou l'autre dimension et qui, une fois validé, constituera un noyau pour développer l'étape suivante. Notons que les étapes de résolution qui suivent sont en ordre inverse des étapes d'analyse précédentes :

*Étape 1 : « Développement d'un algorithme PSO pour la prise en charge de données mixtes »*

Ce travail est en relation avec l'exploration stochastique de la mémoire. Pour rendre l'exploration possible dans un cadre CBR , il faut adapter l'algorithme classique de PSO qui considère uniquement des données à valeurs réelles (ou entières dans les versions discrètes qui ont été proposées) au traitement des données mixtes (continues, discrètes et catégorielles/nominales). Notre contribution est un algorithme capable de fonctionner avec une description de problème comportant des dimensions de nature mixte. Cette étape a été menée à bien et les résultats obtenus sont excellents en comparaison avec d'autres travaux dans le domaine choisi<sup>3</sup>.

---

<sup>3</sup> Voir les résultats détaillés dans :

Nouaouria, N., Boukadoum, M., (2011), A Particle Swarm Optimization approach to mixed attribute data-set classification. IEEE Symposium on Swarm Intelligence, Paris, France, 11 -15 April 2011, pp : 44-51.

Nabila Nouaouria, Mounir Boukadoum, Robert Proulx, (2013) Particle swarm classification: A survey and positioning. Pattern Recognition, Volume 46, Issue 7, July 2013, Pages 2028–2044.

*Étape 2 : « un premier modèle de mémoire basé PSO »*

Ce travail se rapporte aussi à l'exploration stochastique de la mémoire de cas par PSO, mais en relation avec le troisième énoncé de thèse : la notion de structurer l'espace de recherche a priori est remplacée par la recherche de voisinages. L'idée principale est le calcul du voisinage d'un nouveau problème à résoudre par un algorithme PSO (éventuellement mixte) afin de délimiter l'espace de recherche dans la base de cas. Une structure associative reliée à la base permet d'accéder directement aux cas similaires dont la description du problème a été générée dans le voisinage. Pour se faire, une fonction objective représentant une sémantique de similarité est utilisée dans l'algorithme PSO. Cette étape a également été mise en œuvre et les résultats obtenus sont intéressants<sup>4</sup>.

*Étape 3 : « un modèle de mémoire basé PSO intégrant le critère d'adaptabilité »*

Cette étape porte sur la mise en œuvre des solutions pour les deuxième et troisième énoncés de thèse. Après s'être acquitté du devoir de rapidité avec la métaheuristique de PSO, nous avons raffiné le modèle de l'étape 2 avec l'ajout du critère d'adaptabilité comme garant de la qualité du processus de rappel en CBR. Les résultats obtenus pour cette étape sont très prometteurs<sup>5</sup>.

*Étape 4 : « une modélisation CBR dans un cadre applicatif de génération d'itinéraire »*

Cette étape porte sur la représentation du cas à partir d'un cadre applicatif donné à titre d'exemple : la planification de second principe. Notre contribution a été de

---

<sup>4</sup> Voir les résultats dans :

Nouaouria, N., Boukadoum, M., (2010). A Particle Swarm Optimization Approach for the Case Retrieval Stage. *Proc. 30<sup>th</sup> SGAI Int. Conf. on Innovative Techniques and Applications of Artificial Intelligence*. M. Bramer, M. Petridis and A. Hopgood (Eds), p. 209-222, Cambridge, UK, 14-16 December 2010.

<sup>5</sup> Pour illustration de la recherche guidée par la réutilisabilité, voir :

Nouaouria, N., Boukadoum, M., (2010). Case Retrieval with Combined Adaptability and Similarity Criteria: Application to Case Retrieval Nets. *Proc. ICCBR 2010, Alessandria, Italy, July 19-22, 2010*, p. 242-256. In *Case-Based Reasoning. Research and Development*, I. Bichindaritz, S. Montani (Eds.): LNCS 6176 Springer 2010, ISBN 978-3-642-14273-4.

mettre en place les modèles de connaissance nécessaires à la capture de l'expertise de génération de plan, incluant le modèle de connaissances de similarité et d'adaptabilité. L'objectif à atteindre était la prise en charge d'un nouveau cas à résoudre par le processus de recherche. La validation a été faite sur une base de cas représentative d'itinéraires et à visée essentiellement illustrative; en particulier, la qualité de couverture de la base n'est pas considérée. Notons que l'accès à une base de plan de taille réelle nous aurait dispensé d'un tel choix, mais l'indisponibilité des données dans ce domaine nous y a contraints. Le modèle ainsi défini se caractérise par une généralité qui permettra de rester indépendant du domaine d'application (et de sa complexité), tout en s'intéressant à la tâche de résolution de problème qu'est la génération de plan. Secondairement, elle garantira la réutilisabilité du modèle dans d'autres domaines d'application tels que : routage dans les réseaux, robotique...

#### 0.4 Plan du document

La thèse s'articule autour de trois grandes parties, précédées par une introduction générale et suivies d'une conclusion générale. Chaque partie est structurée en deux chapitres abordant chacun un niveau de description différent.

L'introduction générale situe le problème et présente les énoncés constitutifs de la thèse avancée. Elle met l'accent sur les contributions avant de présenter la structure du travail.

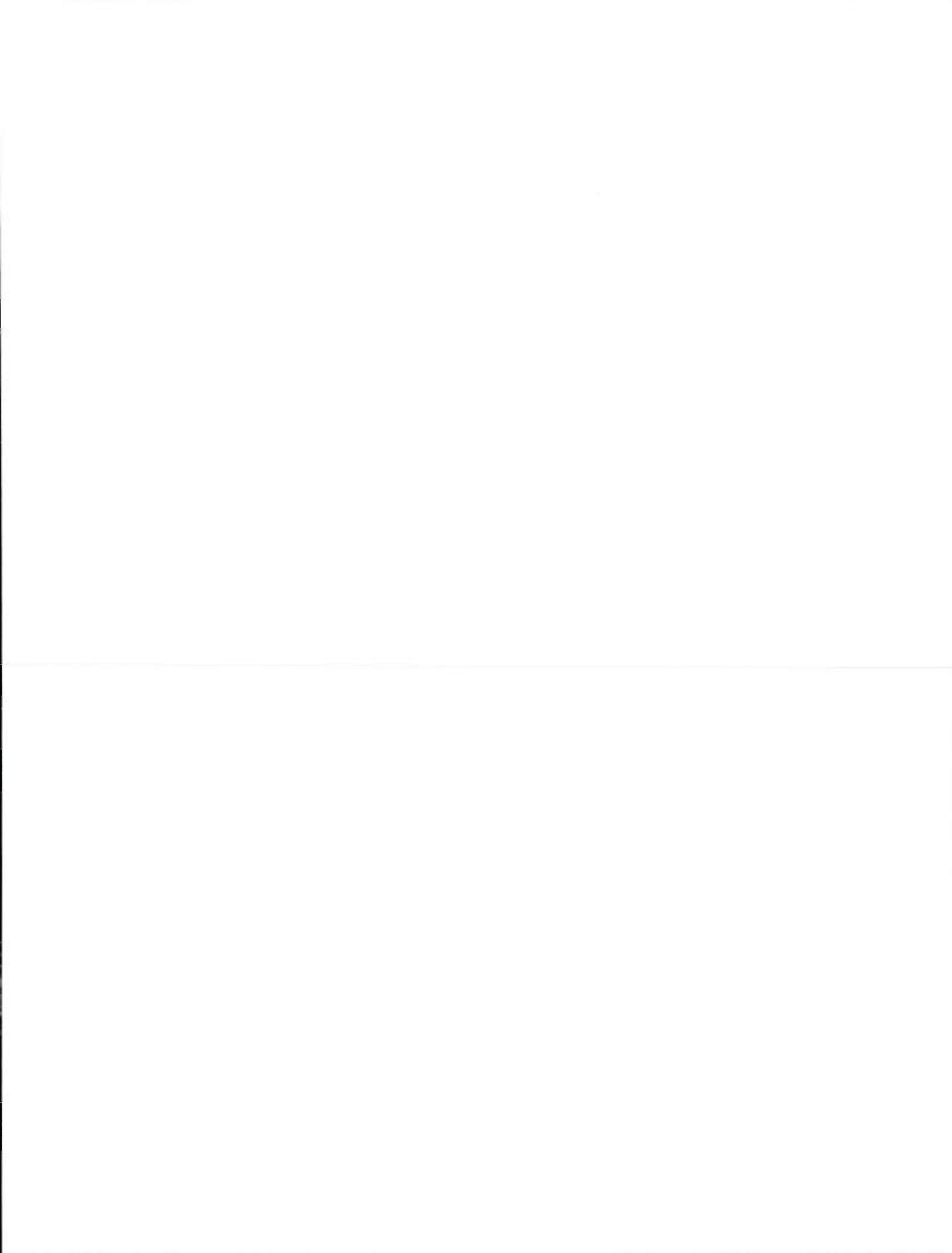
La partie 1 présente les fondements théoriques des paradigmes sous-jacents à cette thèse. Le chapitre I porte sur le raisonnement basé cas (CBR) et de son application à la tâche de résolution de problème qu'est la planification. Le chapitre II, quant à lui, présente les concepts de base de l'optimisation par essaim de particules (PSO). Pour chacun des chapitres, un positionnement du point de vue cognitif est aussi présenté.

La partie 2 porte sur la problématique et la thèse avancée dans ce travail. On retrouve dans cette partie une description graduelle allant du niveau le plus complexe au niveau le plus simple. Elle se subdivise en deux chapitres éclairant chacun une facette du problème. Le chapitre III donne la modélisation de la mémoire basée PSO

sur un plan méthodologique et le chapitre IV illustre l'utilisation de ce modèle à travers un domaine d'application particulier, en l'occurrence, la génération d'itinéraire.

La partie 3 aborde les applications ciblées pour les solutions proposées. Elle se scinde aussi en deux chapitres; le protocole expérimental est présenté et motivé au chapitre V et les résultats des applications illustratives sont présentés puis discutés au chapitre VI.

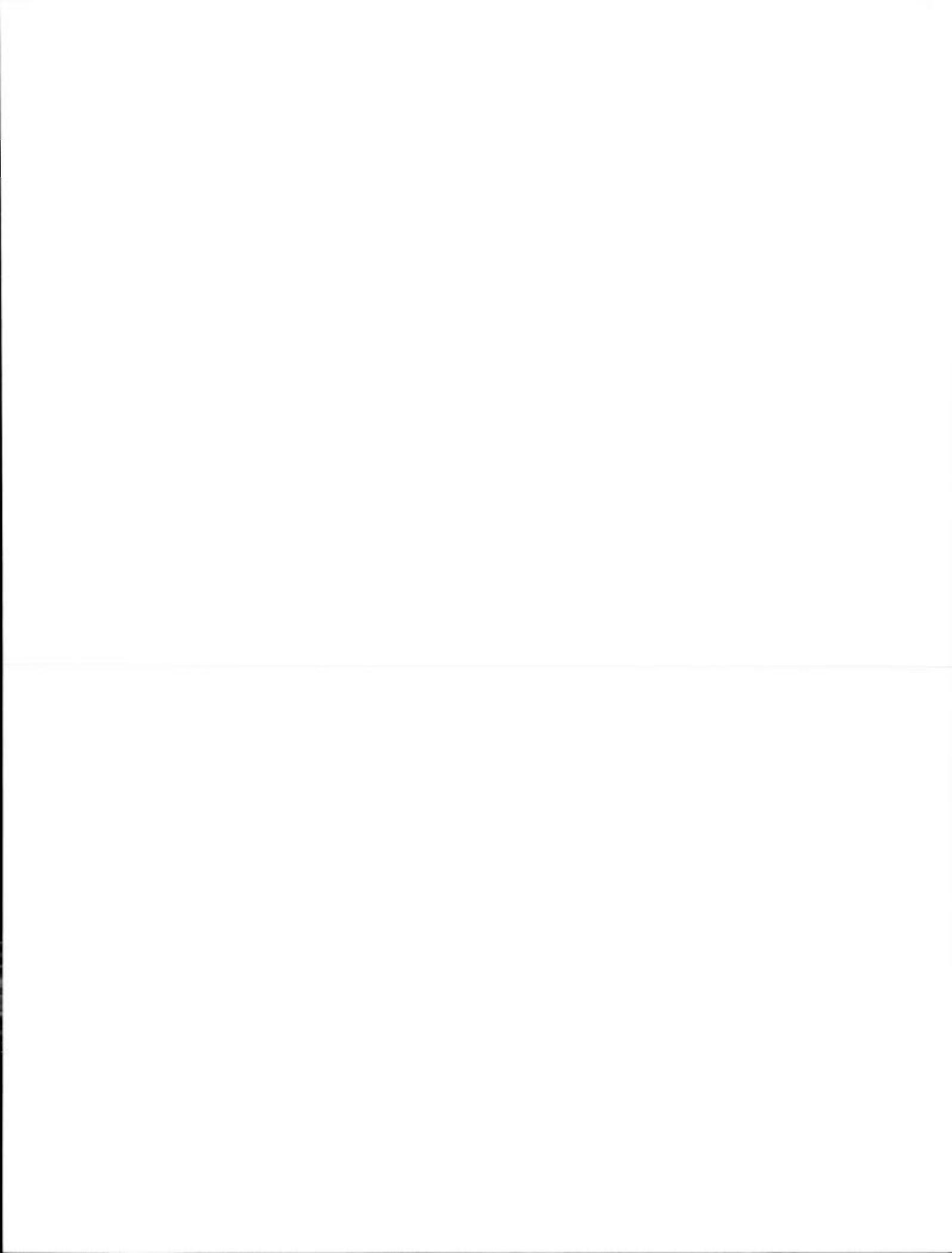
La conclusion générale revient sur le problème posé. Elle donne une évaluation des solutions proposées en termes de contributions avant d'ouvrir les horizons vers des orientations potentielles dans le futur.



## CHAPITRE I

# Le Raisonnement Basé Cas (CBR)

1. Introduction
2. Présentation générale
3. Omniprésence du CBR
4. Modélisation des connaissances en CBR
5. Méthodologie de construction des systèmes CBR
6. Situation du CBR
7. La facette cognitive du CBR
8. La génération de plan basée cas
9. Conclusion



## 1.1 Introduction

Le raisonnement à base de cas est une approche de résolution de problème basée sur la recherche et l'adaptation de 'cas', ou de descriptions épisodiques des problèmes et des solutions qui leurs sont associées (Aamodt et Plaza, 1994). Un système CBR utilise un ou plusieurs cas déjà rencontrés pour le traitement d'un nouveau cas qui se présente. L'ensemble des cas disponibles est emmagasiné dans une base de connaissances qui est enrichie chaque fois qu'un nouveau cas est traité. La base de cas (ou base de connaissances de cas) est assimilable à une mémoire qui est organisée de sorte à rendre possible la comparaison entre le nouveau cas et les cas mémorisés.

L'objectif de ce chapitre est de donner un aperçu général de ce mode de résolution de problème.

## 1.2 Présentation générale

Historiquement, c'est d'abord aux États-Unis qu'émergea le raisonnement à base de cas (Aamodt et Plaza, 1994). La plupart des fondements de l'approche CBR ont été réalisés par Roger Schank et ses étudiants de l'université de Yale dans les années 80. Les travaux de Roger Schank portaient sur la mémoire dynamique et l'apprentissage pour la compréhension d'histoires énoncées en langage naturel. Un des premiers systèmes CBR, CYRUS, développé par Janet Kolodner est un système qui implante les idées de Schank, c'est-à-dire une structure de mémoire dynamique pour la recherche d'informations. De nombreux systèmes furent ensuite développés selon le modèle de Schank pour des tâches différentes, par exemple CHEF ou JUDGE, ou selon d'autres modèles, par exemple un modèle de prototypes comme PROTOS. Durant la fin de cette décennie, la Defense Advanced Research Projects Agency (DARPA) démarra l'application des techniques CBR, ce qui attira l'attention

d'organisations intéressées par les applications commerciales se rapportant aux systèmes à base de connaissance.

Le premier *workshop* sur le sujet eut lieu en 1988. D'autres suivirent en 1989 et en 1991. Cet axe de recherche fait aussi l'objet de *workshop* dans la conférence nationale de l'AAAI (American Association of Artificial Intelligence). En Europe, les premiers travaux datent de 1989. Le premier *workshop* eut lieu à Kaiserslautern, en Allemagne en 1993. Suivirent ensuite de multiples projets importants sur ce thème.

Il existe deux styles du CBR (Ashley et Rissland, 1988 ; Rissland et Skalak, 1989; Kolodner, 1991 ; Kolodner, 1992 ; Lopez de Mantara et al, 2005) : le style résolution de problème et le style interprétation. Dans la résolution de problèmes, on adapte d'anciennes solutions à de nouveaux problèmes : les anciennes solutions peuvent fournir des solutions quasi correctes ou prévenir contre des erreurs potentielles et des échecs. Dans ce style, le CBR peut supporter une variété de tâches de résolution de problèmes telles que planification, diagnostic, conception, etc. Une solution au nouveau problème est proposée sur la base de celle du cas extrait. Elle inclut l'adaptation ou ajustement de l'ancienne solution à la nouvelle situation et une étape de critique ou évaluation de la nouvelle solution.

Dans le style interprétatif, les cas sont utilisés pour évaluer ou justifier de nouvelles situations, ex : un juriste utilisant des cas précédents comme argument pour classer de nouvelles situations. Il y a évaluation même lorsqu'aucune méthode claire et tranchante n'est disponible. On peut aussi interpréter des situations dont les frontières sont floues. En style interprétatif, une interprétation de base (ou un résultat désiré) est proposée, parfois basée sur les extraits, parfois imposée par l'extérieur (ex : lorsqu'un client exige de son avocat un certain résultat). Cette étape est suivie par la justification, processus de création d'un argument pour la solution proposée; ce qui est réalisé par comparaison et opposition de la nouvelle situation aux cas antérieurs. Et enfin, la critique : processus de mise au point, réalisé par la génération

de situations hypothétiques et le test de l'argument sur ces situations. Ces étapes sont dans un sens récursives. La critique et l'adaptation par exemple, requièrent l'extraction de nouveaux cas. Par ailleurs, il y a plusieurs boucles dans le processus. La critique peut mener à une adaptation additionnelle et peut être même à une réévaluation. De plus, lorsque le raisonnement ne progresse pas en utilisant un cas, il peut être réinitialisé entièrement avec un nouveau cas.

### 1.3 Omniprésence du CBR

Le raisonnement à base de cas est un sujet pluridisciplinaire qui met l'accent sur la réutilisation d'expériences. Il est difficile de trouver un consensus sur des définitions plus détaillées du CBR, car il signifie différentes choses pour différents groupes de personnes, comme c'est le cas pour les trois groupes suivants :

- pour la communauté des sciences cognitives, le CBR est un modèle plausible de haut niveau pour le traitement cognitif (Kolodner, 1993).
- les chercheurs en intelligence artificielle considèrent le CBR comme un paradigme de calcul pour la résolution de problèmes (Aamodt et Plaza, 1994).
- Pour les programmeurs de systèmes experts, le CBR est un modèle de conception pour les systèmes experts qui peut être utilisé soit de manière autonome ou intégré à des architectures complexes (Watson, 1997).

L'omniprésence du raisonnement basé cas vient du fait que ses notions de base telles que le stockage, la similarité et l'indexation sont connues des chercheurs de plusieurs disciplines. Ainsi, les chercheurs en CBR sont dans une position unique pour tirer des leçons de progrès dans ces disciplines. En outre, il existe une caractéristique distinctive du CBR, la résolution paresseuse de problèmes (« lazy problem solving ») qui permet de mettre le CBR en avant de la scène au moment de choisir une approche de résolution de problème<sup>6</sup>. La résolution paresseuse de

---

<sup>6</sup> Nous reviendrons sur cette caractéristique distinctive en chapitre IV, au moment de justifier le choix de cette méthodologie dans notre cadre applicatif.

problèmes est une démarche où le calcul est effectué sur la base de la demande. Les solveurs de problèmes purement paresseux sont définis comme affichant les trois comportements suivants (Aha, 1998):

- Différer (« Defer ») : Ils ne traitent pas leurs intrants (c'est-à-dire: les données) avant d'avoir reçu des requêtes d'informations.
- Pilotage par les données (« data-driven »): ils répondent à la requête en combinant des renseignements provenant des données stockées.
- Écarter (« Discard ») : Ils écartent tout résultat intermédiaire temporaire créé pendant la résolution de problèmes.

En revanche, les algorithmes de force brute (« eager ») compilent leurs entrées dans une structure intentionnelle de données (et se débarrassent de leurs intrants), répondent aux requêtes à l'aide de cette compilation à priori abstraite, et la conservent pour de futures requêtes.

Dans le contexte des algorithmes d'apprentissage supervisé, on peut citer en exemple le classifieur des k-plus proches voisins comme algorithme paresseux, tandis que des algorithmes qui induisent les arbres de décision (par exemple Quinlan, C4.5) sont à force brute.

#### 1.4 Modélisation des connaissances en CBR

Dans (Aamodt et Plaza, 1994), le cadre descriptif du CBR a été présenté en deux parties : un modèle de quatre processus du cycle CBR (voir figure 1.1), puis une structure hiérarchique des tâches du CBR.

Le cycle de base du CBR est à quatre phases : «*Retrieve, Reuse, Revise and Retain*». Le système cherche les cas qui s'unifient le mieux à la description fournie par la requête. Ces cas sont stockés dans une base de cas. Si la mémoire de cas est organisée selon une structure particulière, un algorithme de recherche dans la mémoire des cas décrit un parcours dans cette structure. Une phase de filtrage ou sélection est souvent effectuée lorsque l'on dispose d'indices parfaitement

discriminants permettant d'éliminer un sous-ensemble des cas de la mémoire (Kolodner 1993). Une mesure de similarité peut ensuite être utile pour mesurer précisément la ressemblance entre le cas courant et les cas sélectionnés et donner lieu à un classement de ces cas. Le système utilise le problème courant et le cas qui s'y unifie pour générer une solution à ce problème. L'adaptation constitue le deuxième point difficile (après l'indexation) lors de la conception d'un système CBR. Il faut en effet décider quel type de connaissances il est intéressant de transférer du meilleur cas remémoré. On peut effectuer une analogie transformationnelle consistant à transformer la solution du cas remémoré pour l'adapter au cas courant ou procéder par dérivation en adaptant la méthode de génération de la solution.

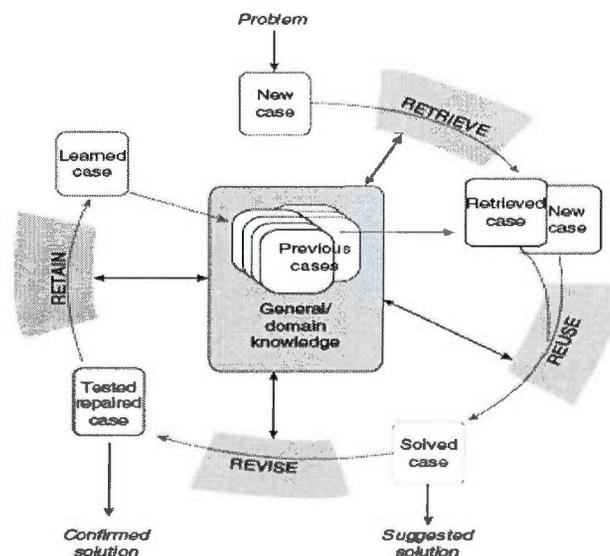


Figure 1.1 Cycle CBR proposé dans (Aamodt et Plaza, 1994)

Par ailleurs, la possibilité d'adapter plusieurs cas pour résoudre un problème de façon simultanée, ou en opérant plusieurs remémorations et adaptations simples aux différentes étapes de la résolution a été jugée plus créative. Par la suite, on procède éventuellement à l'évaluation de la solution proposée en la testant dans un environnement réel ou simulé. Le retour d'information, suite au test, peut alors

guider, en cas d'échec de la solution proposée, un processus de réparation. Le cas échéant, la solution validée est rajoutée à la base de cas pour une utilisation future. La configuration habituelle des systèmes CBR consiste à ranger systématiquement les cas en mémoire. Une mémorisation plus sélective est toutefois possible et utiliserait des critères spécifiques pour juger si le nouveau cas est utile à apprendre vis-à-vis de la mémoire de cas courante. Un cas est utile à apprendre si à partir de son contenu et en utilisant les possibilités d'adaptation, on peut atteindre un point de l'espace des solutions qui était inaccessible avant l'arrivée de ce nouveau cas.

Diverses connaissances sont nécessaires pour l'inférence en CBR: les connaissances permettant de représenter de manière adéquate le nouveau problème à résoudre, les connaissances permettant d'évaluer la ressemblance de deux cas, celles permettant d'organiser la mémoire, ainsi que les connaissances d'adaptation de la solution proposée et les connaissances d'amélioration. La connaissance essentielle reste le cas.

La mémoire est utilisée pendant deux étapes importantes du cycle du CBR : l'étape de remémoration de l'expérience passée pendant la résolution d'un problème et l'étape d'apprentissage ou de mémorisation de la nouvelle expérience acquise en forme de cas.

Par conséquent, le contenu de la mémoire, sa structure, les mécanismes d'indexation utilisés pour assurer l'accès au cas approprié jouent un rôle crucial dans le rappel et l'apprentissage. Retrouver la bonne expérience en mémoire est la clé de succès dans le raisonnement.

#### 1.4.1 Le Cas

Un cas est une entité au sein de laquelle sont rassemblées diverses informations sur une situation passée. Le terme «situation» est très général: description de la conception d'un plan en fonction d'un certain nombre de buts, sentence prononcée pour un délit, etc. Un cas est aussi une entité à partir de laquelle un raisonnement est

possible par un processus consistant à situer la nouvelle situation par rapport aux circonstances définies dans le cas. Schank définit trois types de cas (Rougegrez-Loriette, 1994):

- les cas ossifiés assimilables à une règle générale, qui ne sont associés à aucune expérience vécue, les proverbes en sont un exemple,
- les cas paradigmatiques. Ces cas sont moins généraux que les premiers. Ils sont notamment reliés à un ensemble de situations,
- les histoires, qui constituent des cas uniques. Elles sont très détaillées, ce qui les rend exploitables pour diverses utilisations. Elles constituent la base du raisonnement à partir de cas. Ainsi, une «histoire» est analysée une seule fois, avant la constitution de la base de cas, en fonction de l'objectif du système.

Un cas est constitué de descripteurs, appelés aussi dimensions, répartis en trois : la description du problème, la solution et éventuellement les conséquences de son utilisation (Kolodner, 1993).

La description du problème comprend le contexte dans lequel le cas s'est produit. La solution est la solution du problème décrit dans la première partie ou la réaction à cette description (par exemple, la délibération d'un tribunal, la décision prise, etc.). Elle peut aussi décrire le raisonnement utilisé. L'issue du cas est la description du contexte après la mise en œuvre de la solution.

La description du problème constitue l'état de l'univers lorsque le raisonnement commence. Cela peut être un problème à résoudre ou une situation à interpréter, à classer ou à comprendre. Cette description contient en général (Kolodner, 1993):

- Les buts à atteindre dans la résolution du problème.
- Les contraintes sur ces buts.
- Les caractéristiques de la situation et les relations entre ses parties.

La solution dans un cas peut être de différents types, selon la nature de la tâche à réaliser : une interprétation, une classification, un plan, le produit d'une conception, une explication... Une solution complète contient (Kolodner, 1993):

- La solution proprement dite.
- L'ensemble des pas de raisonnement ayant permis la résolution.
- L'ensemble des justifications pour les décisions prises.
- Les solutions acceptables qui n'ont pas été choisies (solutions alternatives), accompagnées de leurs raisonnements et justifications.
- Les solutions rejetées (avec leurs raisonnements et justifications).
- Les prévisions de ce qui résulterait suite à l'application de la solution.

Il est rare que toutes ces informations figurent dans une solution. Elles sont généralement reportées comme connaissances d'adaptation ou d'évaluation/critiques.

L'issue du cas est la réaction de l'univers à l'application de la solution. On y trouve (Kolodner, 1993):

- Les résultats de l'application de la solution.
- Si l'issue est conforme ou non aux prévisions.
- S'il s'agit d'un échec ou d'une réussite.
- Explication de la conformité ou de la non-conformité aux prévisions.
- Stratégie de réparation.
- Ce qui peut être fait pour éviter le problème.
- Un pointeur vers un autre essai de solution (résultat de la réparation).

Cette partie du cas est généralement omise et les connaissances s'y attachant sont reportées sur les autres étapes du raisonnement.

*A priori*, plus la représentation des cas est riche, plus les cas peuvent être utilisés pour des buts différents. Mais la construction d'une base de cas ne se conçoit pas indépendamment de la tâche que le système doit réaliser.

Les cas doivent être représentés de manière à pouvoir être discernés les uns des autres et ainsi permettre de retrouver le «bon» cas vis-à-vis de l'objectif à atteindre. La détermination des informations utiles aux raisonnements futurs constitue le problème de l'indexation.

Dans ce qui suit, nous reprenons une modélisation formelle du cas (Fuch, 1997) qui tient compte de l'aspect descriptif du problème et de la solution ainsi que des utilisations du cas au travers des tâches de raisonnement.

Un cas noté  $C$  est une connaissance spécifique représentant un épisode de résolution de problème composé de trois parties principales : le problème  $P_b$ , sa solution  $S$  et le raisonnement  $R$  (Figure 1.2).

$$C = (P_b, R, S),$$

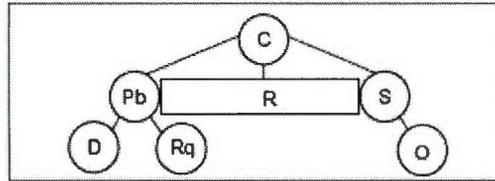
Le problème est décrit par un ensemble de données descriptives  $D$ , et d'une requête  $R_q$  indiquant l'objectif du raisonnement, toujours connu. Les données du problème comportent des contraintes à satisfaire par la solution, et constituant une spécification partielle de  $S$ .

$$P_b = (D, R_q).$$

La solution  $S$  construite par  $R$  doit permettre de satisfaire les objectifs spécifiés par  $P_b$ . La solution à construire par  $R$  peut être un objet quelconque (un plan, par exemple).

Le raisonnement  $R$  est l'ensemble des étapes de raisonnement, les résultats intermédiaires, les décisions prises pour passer de  $P_b$  à  $S$ .

Les connaissances représentées dans les cas sont aussi bien descriptives (Pb,S), qu'opérationnelles ( R ).



**Figure 1.2** Un cas C est composé du problème Pb, du raisonnement R et de la solution S (extraite de (Fuch, 1997)).

Le problème comporte les données D et une requête Rq indiquant l'objectif du raisonnement. La solution S est un objet O construit par R.

L'application de cette définition du cas aux différentes catégories de tâches de résolution permet d'obtenir les définitions suivantes :

- **Planification** : Le problème comprend le descriptif d'un état initial et de l'état final à atteindre par la solution. Le raisonnement construit un plan permettant d'atteindre l'état final à partir de l'état initial.
- **Conception** : la conception est proche de la planification, car il s'agit également d'atteindre un état final exprimé par une spécification partielle de l'objet à concevoir, sous forme de contraintes. Le problème comprend la spécification de l'objet à construire en termes de contraintes. Le raisonnement construit un objet satisfaisant les contraintes à partir des modèles disponibles. La solution peut être un plan ou un autre objet.
- **Diagnostic** : Le problème comprend un ensemble de symptômes et un contexte. La solution est une explication des symptômes constatés.
- **Aide à la décision** : Le problème décrit une situation associée à un contexte, et l'objectif est de découvrir les éléments pertinents en relation avec la situation dont dépend la décision.

- **Interprétation** : Le problème décrit une histoire et l'objectif est de rapprocher l'histoire avec d'autres afin de prédire des évolutions possibles.

Le contenu de la mémoire de cas doit être pertinent pour que le fonctionnement soit le plus performant possible. Plusieurs heuristiques sont définies pour juger le contenu d'une mémoire donnée (Fuch, 1997 ;Smyth et Keane, 1999) : les cas doivent couvrir la tâche de raisonnement à effectuer, ils doivent couvrir les situations de succès les plus communes ainsi que les situations d'échec et ils doivent être collectionnés d'une manière incrémentale pour assurer la couverture du domaine.

Les cas constituant la mémoire doivent être *utiles* et il est préférable de ne pas avoir de redondance dans la mémoire. Le problème de l'*utilité* dans les systèmes d'apprentissage en général apparaît quand la connaissance apprise par le système dans le but d'améliorer sa performance a l'effet inverse, c'est-à-dire provoque une diminution de performance. De manière générale, les cas ne sont pas tous égaux dans la mémoire. Il existe des cas qui agissent sur la compétence du système, et d'autres qui influencent sa performance. Pour modéliser la compétence du système, quatre classes de cas ont été distinguées selon les deux critères suivants (Smyth et McKenna, 1998) : la couverture d'un cas (l'ensemble des problèmes cibles que ce cas résout) et l'accessibilité d'un problème (l'ensemble des cas qui peuvent être utilisés pour générer une solution à ce problème). Les cas sont classés de la manière suivante (Fuch, 1997) :

- **Les cas essentiels** : les cas essentiels sont les cas dont l'effacement réduit directement la compétence du système, ce sont les cas qui ne sont pas accessibles par aucun cas. Ces cas sont généralement isolés et ne peuvent pas être résolus par d'autres.

- **Les cas auxiliaires** : les cas auxiliaires n'influencent pas la compétence du système, un cas est auxiliaire si la couverture qu'il fournit est subsumée par la couverture de l'un de ses cas accessibles.

- Les cas « ponts » les cas ponts n'influencent pas directement la compétence, ils sont nommés ainsi, car leur région de couverture fait la liaison entre des régions qui sont couvertes indépendamment par d'autres cas si des cas de ces régions sont effacés, le cas « pont » devient nécessaire.

- Les cas de support : c'est une classe spéciale des cas « ponts » qui existent en groupe. L'effacement de n'importe quel cas de ce groupe a peu d'influence sur la compétence, par contre l'effacement du groupe entier est semblable à l'effacement d'un cas essentiel.

Cette classification de cas implique un moyen pour ordonner les cas selon leur contribution à la compétence dans le but d'effacer les cas qui diminuent le moins possible cette compétence. Les cas auxiliaires sont les moins importants, ensuite viennent les cas de support, les cas « ponts » et finalement les cas essentiels.

#### 1.4.2 La Base de Cas

Les processus de mémorisation et de remémoration sont fortement liés à la façon d'organiser les cas dans la mémoire. Il est évident que le choix de la structure de mémoire dépend entièrement de la tâche à accomplir, du domaine d'application, de la complexité des étiquettes et du nombre des cas disponibles. Deux catégories pour l'organisation de la mémoire sont distinguées (Kolodner, 1993 ; Malek, 1996) :

**La mémoire plate** : il s'agit de mémoriser tous les cas dans une liste séquentielle. L'avantage d'une telle structure est que pendant la remémoration, tous les cas existant dans la mémoire sont testés; ceci garantit une remémoration précise qui dépend de la qualité de la fonction d'appariement. De plus, la mémorisation n'est pas coûteuse, il suffit d'ajouter le nouveau cas à la fin du fichier. L'inconvénient majeur est le temps de remémoration qui augmente linéairement avec la taille de la mémoire. Il existe plusieurs variantes pour l'organisation en mémoire plate : l'*indexation superficielle* (elle se fait dans un seul niveau), le *partitionnement de la mémoire* (le système doit reconnaître à quelles partitions une nouvelle situation appartient). L'*extraction*

*parallèle* (la fonction d'appariement est appliquée d'une manière parallèle à tous les cas qui existent dans la mémoire).

***La mémoire hiérarchique :*** quand la mémoire des cas est large, il y a une nécessité d'organiser les cas hiérarchiquement. Cela permet de simplifier la remémoration. Il existe globalement, deux approches pour l'organisation hiérarchique des cas en mémoire : les réseaux à traits ou à caractéristiques partagés et les arbres de discriminations.

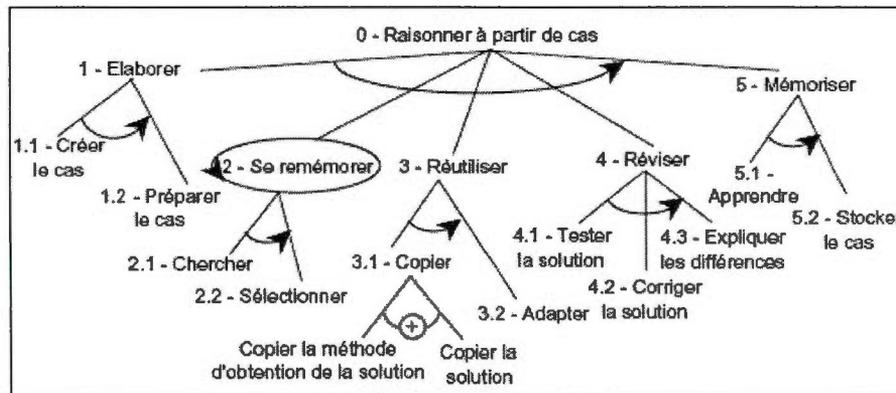
Nous reviendrons sur les aspects techniques de l'organisation de la mémoire au chapitre III dans la partie 2.

#### 1.4.3 Modélisation des Tâches

Le CBR peut être vu sous un autre angle lorsque l'on s'intéresse au flux d'informations inter tâches, les informations utilisées pour contrôler les traitements et les mécanismes qui supportent les tâches. Une tâche est décrite par les informations qu'elle traite, les fonctions réalisées, les informations qu'elle produit, les modèles de connaissances utilisés, et des mécanismes de raisonnement ou d'inférence mis en œuvre. La tâche est activée par une requête formulant un problème à résoudre et fournit en sortie un cas contenant une solution au problème. Elle utilise des modèles conceptuels de connaissances du domaine d'application et une mémoire de cas passés qu'elle enrichit par apprentissage. Elle utilise des mécanismes de classification et d'appariement.

Le modèle de tâches permet de mettre en évidence les décompositions de tâches et leur ordonnancement. Il s'agit d'un formalisme hiérarchique qui décrit une décomposition de tâches en sous tâches. Une décomposition de tâches en sous tâches est donnée en figure 1.3. La tâche principale la plus abstraite est située à la racine de la hiérarchie, et reliée à ses sous tâches situées plus bas. Les arcs matérialisant les relations de composition de tâches sont annotés à l'aide de trois symboles permettant

de préciser la nature de la relation entre la tâche principale et ses sous tâches (Aamodt et Plaza, 1994 ; Fuch, 1997).



**Figure 1.3 La décomposition (incomplète) de la tâche Raisonner à partir de cas (extraite de (Fuch, 1997)).**

La tâche principale Raisonner à partir de cas est réalisée par les sous tâches Elaborer, se remémorer, réutiliser, Réviser puis Mémoriser en séquence. La tâche se remémorer est répétitive, et la tâche Copier peut être réalisée par l'une ou l'autre des sous tâches : Copier la méthode d'obtention de la solution ou Copier la solution. L'absence d'annotation indique un ordre non défini (en séquence ou en parallèle).

#### 1.4.3.1 Élaboration du problème

Avant d'effectuer la remémoration d'un cas, il est nécessaire de créer un nouveau cas, de collecter des informations afin de décrire le problème, de déterminer les indices qui permettront de rechercher un cas dans la mémoire des cas passés. La tâche d'élaboration est déclenchée par un agent extérieur (un utilisateur, un logiciel), et a pour point de départ une requête qui donne les premières informations connues et contextuelles sur le problème à résoudre. Un nouveau cas est créé et la description du problème est ensuite complétée en collectant d'autres informations. Puis sont déterminés les indices à utiliser pour remémorer un cas passé. Notons que dans la

pratique, la plupart des systèmes CBR ne comprennent pas de tâche d'élaboration de manière explicite.

#### 1.4.3.2 Remémoration

La remémoration choisit un cas ayant le plus de chances de faciliter la résolution du problème généralement, un cas passé similaire au problème courant. Cette tâche exploite un modèle d'indexation pour identifier les points d'entrée dans la mémoire des cas, des modèles de similarité et des modèles du domaine afin d'évaluer la ressemblance. La tâche de remémoration procède à la sélection progressive d'un sous-ensemble de cas, en commençant avec la mémoire de cas intégrale jusqu'à obtenir un seul cas dont la solution sera réutilisée pour le problème courant. C'est une tâche à raffinement successif. Chaque étape de raffinement commence par une recherche qui consiste en un appariement suivi d'une évaluation de la similarité. Les cas sont alors ordonnés. La sélection choisit un ou plusieurs cas en fonction de cet ordonnancement. La tâche de recherche peut consister à parcourir une hiérarchie d'index et à comparer chaque nœud avec les indices. S'il y a appariement, le sous-arbre est alors sélectionné et ses descendants sont comparés de la même manière. L'ensemble des cas sélectionnés est raffiné au fur et à mesure de l'évolution dans la hiérarchie. Pour le calcul de similarité, le processus est généralement réalisé en une seule phase. Une autre forme de recherche appelée propagation d'activation consiste, à partir des caractéristiques du nouveau cas, à parcourir des relations vers des connaissances (connaissances du domaine ou connaissances d'adaptation par exemple) et en activant des cas atteints par différentes relations à la fois. Dans la plupart des systèmes CBR, la remémoration est réalisée en deux phases : filtrage (ou discrimination) et évaluation de la similarité des cas sélectionnés.

#### 1.4.3.3 Réutilisation

La solution du cas remémoré est réutilisée pour le nouveau problème moyennant généralement des adaptations. L'adaptation consiste à modifier la solution du cas remémoré pour prendre en compte les différences entre spécifications de problèmes.

Généralement, la solution du cas remémoré est copiée et certaines parties sont substituées en fonction des spécificités du nouveau problème. Beaucoup de systèmes se contentent d'une recopie simple de la solution du cas remémoré ou d'une composition des solutions de plusieurs cas remémorés dans différents cycles de raisonnement distincts. L'adaptation ne change pas radicalement la solution, mais en modifie certaines parties et peut la réorganiser. Les systèmes distinguent généralement l'adaptation par transformation (c'est la solution du cas elle-même qui est réutilisée) et l'adaptation par dérivation (c'est la trace du raisonnement ayant produit la solution du cas qui est réutilisée). La tâche 'adapter' procède à un appariement (lorsque ça n'est pas déjà fait durant la remémoration) puis modifie la solution en se focalisant sur les différences entre spécifications de problèmes pour déterminer quels éléments de solution doivent être modifiés et quelles méthodes d'adaptation appliquer sur la solution.

#### 1.4.3.4 Révision

La solution obtenue après adaptation est proposée. Différentes sources permettent d'obtenir des retours sur le succès ou l'échec de cette solution. D'une part la mise en œuvre de la solution permet de détecter des échecs, et d'autre part sa soumission à un utilisateur lui permet d'apporter des corrections. La révision est souvent considérée comme la continuation de l'adaptation. La tâche d'adaptation est considérée comme l'ensemble des tâches modifiant la solution avant sa mise en œuvre. La tâche de révision prend en compte les retours après la mise en œuvre de la solution permettant de détecter des cas d'échecs non connus auparavant, et prépare à l'apprentissage. La révision permet d'effectuer des corrections de la solution et d'exploiter les différences entre la solution avant et après correction. Les différences observées entre la solution après adaptation et la solution après révision peuvent être considérées comme des échecs de raisonnement. L'identification d'une cause d'échec peut être typiquement réalisée grâce à un processus de diagnostic qui recherche des explications possibles de l'échec constaté en exploitant un modèle causal (s'il existe). L'explication peut

également être simplement demandée à un utilisateur. Les explications sont ensuite exploitées par la tâche d'apprentissage pour acquérir de nouvelles connaissances afin d'éviter des échecs ultérieurs et améliorer les compétences de raisonnement.

#### 1.4.3.5 Mémorisation

Un cas résolu est stocké s'il apporte de nouvelles compétences au système, afin de pouvoir être remémoré ultérieurement. Au fur et à mesure que de nouveaux cas sont rencontrés et résolus, ils sont ajoutés à la mémoire des cas et les index sont mis à jour. La tâche 'mémoriser' comporte la composante d'apprentissage proprement dit du CBR. Les deux tâches d'apprentissage sont l'exploitation des résultats de la tâche de révision afin de modifier les modèles de connaissances pour tenir compte des échecs, et le stockage du cas dans la mémoire. La composante d'apprentissage est fondamentale en CBR, car elle permet au système d'évoluer au fur à mesure que de nouveaux cas sont acquis. L'apprentissage dans le CBR couvre différents aspects. Il consiste d'abord à acquérir de nouveaux cas résolus et à mettre en place les index appropriés pour permettre leur remémoration ultérieure. La mise en place d'index permet de prendre en compte les échecs rencontrés par la solution et les stratégies utilisées pour les corriger. Le principe de base est qu'une erreur commise ne doit pas être répétée, et des index spécifiques sont mis en place afin de remémorer, en même temps que le cas, des informations sur les échecs potentiels.

### 1.5 Méthodes de construction des systèmes CBR

Des méthodologies guidant le développement des systèmes CBR sont apparues. Elles partagent les trois phases suivantes (Allen, 1994 ; Bergmann et Althoff, 1998 ; Bartsch-Spörl et al., 1999):

#### 1.5.1 Conception de la Base de cas

On développe une représentation générale des cas, en utilisant les sources matérielles d'information (documents, enregistrement de bases de données, notes de l'expert...). Ceci étant accompli en coordonnant les efforts entre utilisateur,

gestionnaire et concepteur du système. La conception de la base de cas peut mener à la constitution d'un lexique de termes utilisé pour décrire les caractéristiques du problème, la sélection des caractéristiques appropriées pour l'indexation des cas, la spécification de schémas de bases de données utilisées pour le stockage des cas (ainsi que la définition de standards d'écriture de la base de cas).

#### 1.5.2 Développement initial de la Base de cas

Une base initiale est développée pour fournir les grandes lignes de l'application. Elle sera par la suite, revue en extension par le concepteur et l'utilisateur et itérativement raffinée jusqu'à ce qu'elle couvre une large portion du domaine d'application.

#### 1.5.3 Développement continu et maintenance

La base initiale est utilisée et raffinée à travers les étapes de validation et stockage. Une partie de l'organisation gère la base de cas ainsi que la représentation de cas de la même façon qu'une base de données traditionnelle. Des techniques de contrôle de qualité statistiques peuvent être utilisées pour connaître la précision et l'utilité des cas.

Un environnement typique de développement de CBR fournit: des schémas par défaut de représentation de cas, un flot prédéfini de résolution de problèmes support de décisions. Des formats sont utilisés pour l'édition de cas, de caractéristiques et de solutions. Des utilitaires sont fournis pour l'indexation manuelle et automatique, l'importation automatique de cas à partir des enregistrements d'une table de BD relationnelle ainsi que le groupage conceptuel (conceptuel clustering) de cas pour l'analyse et la recherche efficace.

### 1.6 Situation du paradigme CBR

Le raisonnement à base de cas typique est caractérisé par la richesse de la représentation des cas, l'adaptation du cas trouvé relativement au nouveau contexte de résolution ainsi que l'utilisation de connaissances générales. Le CBR s'apparente ainsi à d'autres paradigmes, soit par ses origines, soit de par son fonctionnement,

mais il s'en distingue aussi. Dans cette section nous le situons dans le paysage des paradigmes:

- le raisonnement basé exemplaires où l'abstraction d'un problème est définie par extension, à travers ses exemplaires. Dans cette approche, la résolution du problème est essentiellement une tâche de classification des exemplaires.
- le raisonnement basé instances qui est une spécialisation du précédent et est considéré comme une approche CBR hautement syntaxique.
- Le raisonnement basé mémoire qui considère la collection de cas comme étant une large mémoire et le raisonnement comme étant un processus de recherche et d'accès à cette mémoire. L'utilisation du traitement parallèle caractérise cette approche.
- Le raisonnement basé analogie est caractérisé par la résolution de nouveaux problèmes en réutilisant d'anciens cas appartenant à des domaines différents. Il s'agit essentiellement de l'application de mécanismes d'identification et d'utilisation dans le cadre d'une analogie interdomaine alors que le CBR est vu comme une analogie intradomaine.

Le CBR entretient certes, d'étroites relations avec ces concepts, mais il ne peut être confondu avec l'un des deux sans connaître les limites de chacun.

#### 1.6.1 CBR et Analogie

Bien que le raisonnement par analogie fût à l'origine du CBR, ce dernier s'en est distingué par le développement de techniques de remémoration inspirées des théories de la mémoire. En fait, le CBR peut être considéré comme une forme particulière du raisonnement par analogie. Ce dernier infère des connaissances sur une entité appartenant à un univers cible en important des connaissances sur une entité appartenant à un univers source. En CBR, les univers source et cible sont confondus. Un exemple illustrant parfaitement cette situation a été cité par Kodratoff (1993) :

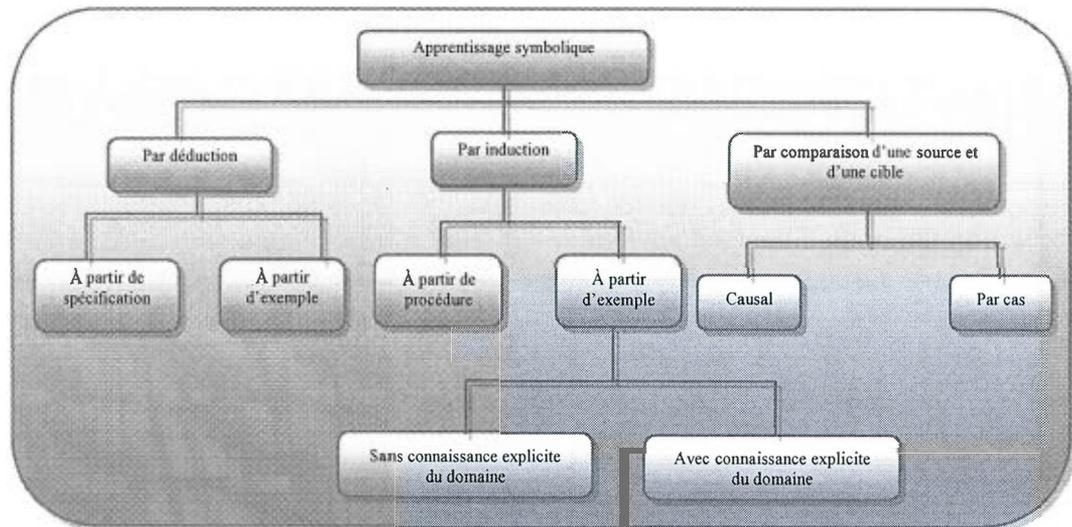
Roméo disant à Juliette qu'elle est son «soleil», détecte une similarité entre le soleil et Juliette, en l'occurrence, que le soleil est un objet astronomique qui conditionne la vie sur les planètes qu'il attire, et que Juliette est un objet humain qui conditionne la vie de Roméo qu'elle attire. Les propriétés physiques du soleil correspondent également à des propriétés psychologiques de Juliette. Il s'agit là d'une analogie faite entre le soleil et Juliette.

En raisonnement basé cas, le concept de Juliette est très éloigné du concept soleil (ils ne sont pas de même nature et n'appartiennent pas au même univers), on ne peut donc comparer leur cas. Mais Juliette est très proche de Marguerite dans Faust (deux jeunes filles innocentes, etc...) et on peut essayer d'appliquer le cas Juliette à celui de Marguerite. On constate là l'importance d'un univers unique pour le raisonnement basé cas.

C'est ce qui fait du CBR une variante simple du raisonnement par analogie, ce dernier étant naturel, mais difficile à mettre en œuvre dans la pratique (Haton, 1993).

#### 1.6.2 CBR et Apprentissage automatique

La flexibilité et la possibilité d'acquisition incrémentale de connaissances épisodiques font des systèmes CBR des systèmes capables d'évolution. Cette évolution peut être qualifiée d'apprentissage «paresseux» par opposition à l'apprentissage «agressif» occasionné par certaines méthodes d'apprentissage inductif (Smail, 1994). En effet, l'apprentissage à base de cas se caractérise par une abstraction «paresseuse» de l'expérience antérieure. Contrairement à une abstraction «agressive» ou une compilation sévère de l'expérience en prévision d'un usage futur, cette méthode préserve le maximum d'informations sur les cas traités jusqu'à ce qu'une utilisation réelle se présente. Kodratoff (1993) a établi une taxonomie des techniques de raisonnement utilisées pour effectuer l'apprentissage (voir figure 1.4).

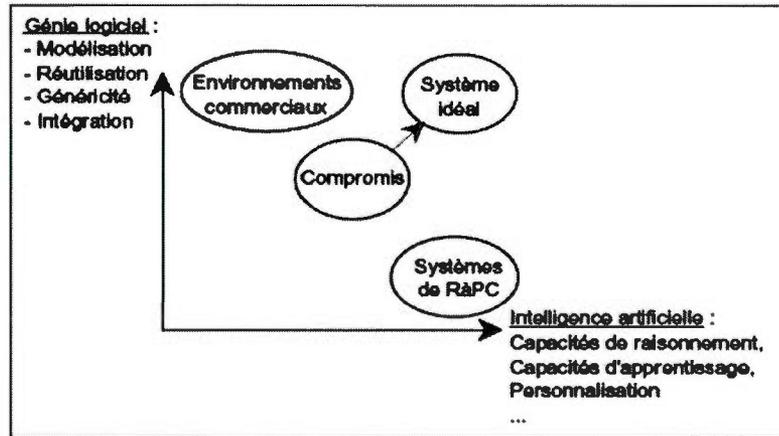


**Figure 1.4 Techniques de raisonnement pour l'apprentissage (extraite de (Kodratoff, 1993)).**

Il situe l'apprentissage par cas au même niveau que l'apprentissage causal, puisque tous les deux sont basés sur la comparaison d'une source et d'une cible avec la différence que, pour l'apprentissage causal, on recherche des relations de causalité alors que le raisonnement par cas recherche des relations de proximité.

### 1.6.3 CBR et Réutilisation

La réutilisation est souvent systématiquement associée au génie logiciel. On la définit comme étant l'application des connaissances d'un système, déjà opérationnel à un autre système similaire (Erzan et al, 1999). D'un point de vue CBR, on note la place centrale qu'occupe ce concept dans le cycle de base : <Retrieve, Reuse, Store>, mais il s'agit là d'une réutilisation à sens plus large que celui du génie logiciel. On réutilise un épisode passé pour résoudre un problème actuel. Cet épisode constitue une leçon apprise lors du processus de résolution avec tout ce que cela inclut comme connaissance opératoire ou factuelle. En fait, cette forme de réutilisation est inhérente au processus d'importation de connaissances d'une entité source vers une entité cible que l'on retrouve aussi bien en raisonnement par analogie que par cas.



**Figure 1.5** Les axes génie logiciel/intelligence artificielle (extraite de (Erzan et al, 1999))

Dans la figure 1.5, on voit que les paradigmes de l'intelligence artificielle et génie logiciel sont distincts et caractérisent des objectifs différents, souvent contradictoires. Pour l'aspect génie logiciel, il s'agit de fournir un environnement méthodologique pour guider la conception et des outils associés facilitant la réutilisation de fonctionnalités fournies. Les outils commerciaux sont dédiés à la conception et la réalisation d'applications utilisant le CBR et doivent répondre plus à des objectifs de génie logiciel dans le but de fournir un outil directement réutilisable dans différents domaines d'application. En ce qui concerne l'aspect intelligence artificielle, il s'agit d'obtenir de bonnes capacités de raisonnement, de faciliter l'acquisition de connaissances et l'apprentissage. Ceci nécessite une définition explicite des connaissances, ce qui rend les systèmes plus souples quant à l'ajout de fonctionnalités. Les systèmes CBR reposant sur un environnement de représentation de connaissances permettent plus facilement l'implantation des modèles des connaissances de manière explicite. Ils sont par conséquent potentiellement plus ouverts pour une personnalisation à des domaines d'application particuliers. Néanmoins, la personnalisation reste difficile si un cadre méthodologique adéquat ne guide pas la conception.

Le système idéal répondant à la fois aux deux objectifs, génie logiciel et puissance de raisonnement et d'apprentissage, n'est pas évident à obtenir (dans la mesure où certains objectifs présents dans l'un des axes sont contradictoires avec ceux du second axe comme c'est le cas pour la généralité et la personnalisation). Néanmoins, moyennant un bon compromis entre ces deux composantes, on se rapprocherait de l'optimum.

### 1.7 La facette cognitive du Raisonnement basé cas

Aha (1998) a énuméré une liste de thèmes dans diverses disciplines liés à la résolution paresseuse de problèmes (Tableau 1.1). Tous se trouvent avoir un lien direct ou indirect avec le CBR.

**Tableau 1.1 Quelques domaines de recherche liés à la résolution paresseuse de problème (traduit de (Aha, 1998)).**

Domaine de recherche	Thème
Psychologie Cognitive	Modèles basés exemples
Reconnaissance des formes	Classifieurs basés sur les K-plus proches voisins
Apprentissage Machine	Plus que les approches basées instances
Cognitive Science	CBR, Analogie
Recherche d'information	Recherche de documents
Statistique, Robotique	Régression pondérées localement
Structures de données	Variante récente des k-d tree
Génie Logiciel	Réutilisation logicielle
Planification de processus	Planification de processus variant

Par ailleurs, dans leur article : « Quelques points de repère sur l'évolution de l'étude du raisonnement en psychologie cognitive », Ripoll et Tricot (1995) terminent par des conclusions jugées révolutionnaires à notre sens. En effet, elles remettent en question des siècles de pensée logique en psychologie cognitive et des décennies de raisonnement basé règles en Intelligence Artificielle en propulsant la mémoire en

avant de la scène du Raisonnement. Dans ce qui suit, nous reprenons l'essentiel de leurs conclusions :

Pour les chercheurs en psychologie cognitive comme pour les pédagogues, il existait une croyance bien établie: l'idéal de la connaissance serait une connaissance abstraite et parfaitement décontextualisée, susceptible de s'appliquer quel que soit le domaine considéré. Les mathématiques et la logique sont considérées comme des disciplines majeures, précisément parce qu'elles impliquent un niveau d'abstraction maximum: elles tendent vers l'universalité. En fait, cette croyance dans le caractère essentiel de la pensée inférentielle déductive découle d'une autre croyance, encore plus fondamentale. Il s'agit de la croyance selon laquelle l'essentiel de l'activité cognitive consiste à raisonner et à résoudre des problèmes. Dès lors, l'objectif essentiel de la recherche a consisté à décrire, à formaliser puis à modéliser, les règles, les heuristiques et les algorithmes qui régissent notre faculté de penser et de raisonner. Mais les résultats de cette entreprise, commune à l'Intelligence Artificielle et à la psychologie, n'ont pas été à la hauteur des attentes. D'une part, de tels modèles n'approchent que de très loin les capacités cognitives des humains, d'autre part, de nombreuses fonctions cognitives demeurent hors d'atteinte de tels modèles (l'induction, la reconnaissance de formes, la décision en situation d'incertitude,...), enfin et surtout de nombreuses données empiriques ont cruellement parlé en défaveur de cette première conception de "Raisonner".

[...] Sous la pression de données empiriques problématiques, un renversement théorique profond s'est amorcé et se développe actuellement. Ce renversement théorique a au moins deux facettes :

- La pensée logique est l'exception, l'inférence non démonstrative, plutôt la règle.
- L'homme est un piteux "raisonneur" (ou calculateur), mais un prodigieux "activateur".

Ces deux points ont été clairement mis en évidence à partir d'études sur les joueurs d'échec. Les grands maîtres raisonnent finalement assez peu, ils n'envisagent et ne traitent en profondeur qu'une infime partie des déplacements de pièces qu'autorisent les règles du jeu. En fait, ils parviennent, avec une rapidité incroyable, à retrouver dans leur mémoire les coups pertinents. Leur expertise semble davantage résulter d'une capacité mnésique remarquable que d'une capacité à raisonner, au sens classique qu'on donne à ce terme. Face à ce constat, l'étude du raisonnement a évolué dans plusieurs directions plutôt complémentaires qu'opposées.

[...] L'étude de la mémoire devient un thème central dont on ne peut plus faire l'économie, quand bien même on ne s'intéresserait qu'au Raisonnement (avec

un grand R). Globalement, il semble bien que pour "bien raisonner", l'essentiel est d'activer les connaissances pertinentes.

Les renversements théoriques décrits dans (Rippol et Tricot, 1995) suggèrent que les connaissances générales et abstraites ne s'enseignent pas, elles se construisent activement. La manière dont on les exploite dépend en grande partie de la manière dont on les a construites, ce qui en retour détermine les conditions d'applications de telles connaissances. Apprendre à raisonner implique peut-être davantage d'apprendre à organiser sa mémoire de telle sorte que les informations pertinentes soient facilement accessibles que d'apprendre à aligner des raisonnements de type logique.

Ces Conclusions introduisent et argumentent de manière intuitive, d'une part l'intérêt du raisonnement à base de cas (CBR pour Case Based Reasoning) lorsqu'elles abordent les questions « d'inférence non démonstrative » et « de transfert d'apprentissage » ; d'autre part, elles mettent en avant « la mémoire » comme thème essentiel d'étude pour le raisonnement.

Notons par ailleurs que les origines du raisonnement basé cas sont profondément ancrées dans la psychologie cognitive pour l'étude de la mémoire (Rougegrez-Loriette, 1994). En effet le raisonnement à partir de cas complète le raisonnement par analogie par un mécanisme de mémorisation et d'extraction des expériences.

La résolution de problème dans un style CBR fait écho à la position radicale de Minsky (Caplat, 2002) selon laquelle l'approche logique ne peut fonctionner en I.A. En résumant son point de vue, Minsky déclare que le raisonnement logique (déductif) n'est pas assez flexible pour servir de base à la pensée, que la rigueur imposée par la logique formelle n'est pas souhaitable, et qu'en conséquence, les systèmes conçus sont beaucoup trop rigides. Il leur préfère la notion de « cadres » (*frames*) qui sont des entités regroupant de façon structurée l'ensemble des informations relatives à un objet ou une situation stéréotypiques. Les « cadres » sont reliés entre eux en réseaux sémantiques et c'est ce réseau qui sera exploité/parcouru pour trouver le « cadre » le plus adapté à une situation observée. Quand un « cadre » est proposé, un processus de

mise en correspondance (*pattern matching*) remplit les informations manquantes. Des facettes particulières (exceptions, valeurs par défaut) complètent le dispositif.

Dans la même mouvance, on peut citer les « scripts », introduits par Schank et Abelson (Kolodner, 1993), dont les travaux portaient sur la compréhension de texte énoncé en langage naturel. L'hypothèse fondamentale est que la compréhension d'histoire s'appuie sur des schémas mentaux qui comblent les « vides présents » naturellement dans les textes. Un « script » décrit un épisode selon un comportement conventionnel découpé en événements qui se succèdent selon un ordonnancement décrit par l'ordre d'apparition attendu de ces événements. Il contient un certain nombre de facettes (but poursuivi, plan relation sociale, rôle tenu... et de manière générale tout ce qui peut jouer sur le comportement dans la mise en œuvre du schéma). Tout comme les « frames », ces « scripts » sont unifiés à la situation courante.

Plusieurs théories de la mémoire ont successivement dominé. La dernière étant la théorie de la mémoire dynamique de Schanck. Elle a donné lieu aux premiers systèmes de raisonnement basés cas (Rougegrez-Loriette, 1994 ; Kolodner, 1993). Ce qu'il y a lieu de préciser est que la théorie de la mémoire dynamique fournit un guide pour la représentation, la segmentation, l'accès et le raisonnement sur les « cas ».

### 1.8 La génération de plan en CBR

Parmi les activités cognitives les plus usuelles et qui ont été mimées par l'intelligence artificielle (IA), on retrouve la planification. L'IA cherche à élaborer des méthodes de représentation de connaissances et des algorithmes permettant de les manipuler puis à les implémenter dans des *planificateurs* afin de reproduire et même d'améliorer ce processus.

L'objectif de cette section est de donner un aperçu général de la planification basée cas dans le cadre du calcul d'itinéraire avec préférence. L'idée est de nous

doter du bagage théorique nécessaire au développement d'une application dans le domaine.

L'une des méthodes d'accélération de la planification en IA consiste à se servir de fragments de plans déjà élaborés pour en construire un nouveau. Cette méthode a été employée pour la première fois d'une manière élaborée dans STRIPS (Régnier et Vidal, 2004). Lorsqu'un plan est construit, le planificateur le mémorise sous la forme d'un macroopérateur généralisé. Pendant la planification, STRIPS passe en revue les macroopérateurs pour en trouver un qui soit applicable et qui permette d'obtenir certains des buts recherchés. Ce macroopérateur est alors simplifié de manière à ne conserver que les opérateurs utiles à l'obtention des buts désirés, puis intégré dans le plan. Cette méthode permet ainsi d'obtenir des plans corrects avec un effort de planification moins important. On retrouve une approche similaire dans la planification à base de réutilisation de plans (case-based planning): le système choisit un des plans dont il dispose (méthode dépendante du domaine) et cherche à le modifier de manière à ce qu'il s'adapte à la nouvelle situation.

L'approche générative classique utilisée pour le processus de planification consiste globalement en une recherche à travers l'espace des opérateurs possibles pour la résolution du problème. Pour la majeure partie des problèmes d'ordre pratique, cette recherche est insoluble (Bergmann et al., 1998). Le CBR, peut être un bon palliatif dans la mesure où il transfère des solutions déjà existantes plutôt que de chercher de nouvelles. D'où l'émergence d'une sous-discipline du CBR, appelée CBP (pour Case Based Planning).

La planification basée cas est une planification basée sur le rappel en mémoire (Hammond, 1989). Elle est basée sur la réutilisation d'expériences passées, à savoir la réutilisation de plans qui ont déjà réussi lors de situations passées et la récupération des plans qui ont échoué. Ce paradigme couvre une gamme de stratégies différentes pour organiser et gérer les anciens et nouveaux plans, par exemple : les plans peuvent

être retenus en tant qu'expériences concrètes ou en tant que traces de génération (Spalazzi, 2001).

Le CBP hérite de quelques-unes des motivations du CBR telle que la plausibilité psychologique. Par contre, l'absence de cadres formels ne semble pas être une motivation récurrente pour la planification basée les cas. La plupart des domaines d'application de planification ont une structure et le processus de résolution de problèmes peut être formalisé avec des théories bien fondées (ex : le transport logistique et la planification de processus de fabrication). Néanmoins, il existe certains domaines où ce n'est pas le cas (ex : en médecine, l'expérience peut être le seul soutien pour planifier analyses et traitements).

Une autre motivation est l'efficacité. Certains auteurs affirment que la capacité de réutiliser des parties d'anciens plans améliore l'efficacité de la planification en évitant la répétition des mêmes efforts. Malheureusement la vérité de cette déclaration est controversée. D'un point de vue théorique l'analyse faite par Nebel et Koehler (1995), montre que dans la réutilisation des plans est, dans le pire des cas au moins, aussi complexe que la génération de plan de premier principe. D'un point de vue expérimental il ya des résultats contradictoires reportés dans (Spalazzi, 2001) : d'un côté il ya les expériences qui semblent suggérer que la planification basée cas est toujours mieux que la planification générative. Dans d'autres études empiriques, il est mentionné qu'un gain d'efficacité réelle est fortement dépendant du domaine et du degré de similarité des cas. De manière générale, Koheler (1994) avance la thèse selon laquelle la réutilisation de plan mène à un gain de performance lorsque l'inéquation 1 est vraie.

$$t(\text{retrieval}) + t(\text{matching}) + t(\text{refitting}) < t(\text{planning from scratch}) \quad (0)$$

L'effort de réutilisation exprimé ici en temps doit être comparé à l'effort induit par une technique générative. Nous reviendrons sur cette inéquation dans la partie 2 au chapitre III.

### 1.8.1 Modèle de connaissances et cycle d'inférence en CBP

Une des questions majeures en CBP est la représentation des plans et leur organisation en mémoire. Fondamentalement, il s'agit de décider quoi stocker et comment sera organisée la mémoire afin de récupérer et de réutiliser les anciens plans de manière efficace. L'efficacité et l'efficacé d'un planificateur CBP sont fortement tributaires de sa représentation du plan et de son organisation de la mémoire. Ceci est une conséquence naturelle du fait que sa résolution de problèmes soit essentiellement basée sur la remémoration et l'adaptation d'anciens plans.

Un problème de planification est formalisable (Spalazzi, 2001) par un nuplet  $\langle I, G, Op \rangle$  avec :

$I$  : ensemble de formules décrivant l'état initial,

$G$  : ensemble de formules décrivant la spécification du but,

$Op$  : ensemble d'opérateurs ou actions utilisables dans le plan.

Toute action  $a \in Op$  est décrite par ses pré-conditions  $C_a$  et ses post-conditions ou effets  $E_a$ .

Un plan  $\Pi$  est un nuplet  $\langle S, B, O, L \rangle$  où

$S$  : ensemble des actions du plan ayant deux actions particulières :  $t_I$  qui a comme effets  $I$  et  $t_G$  qui a comme pré-conditions  $G$ .

$B$  : est un ensemble de liens autorisés et interdits sur les variables apparaissant dans  $\Pi$ .

$O$  : est une relation d'ordre sur  $S$ , notons que  $t_I$  est toujours la première action et  $t_G$  la dernière action. Lorsque la relation d'ordre est totale,  $\Pi$  est appelé plan linéaire sinon, il est appelé non-linéaire.

$L$  : est un ensemble de liens causaux de la forme où  $(s \xrightarrow{p} s')$   $s, s' \in S$  et  $p \in E_s$  et

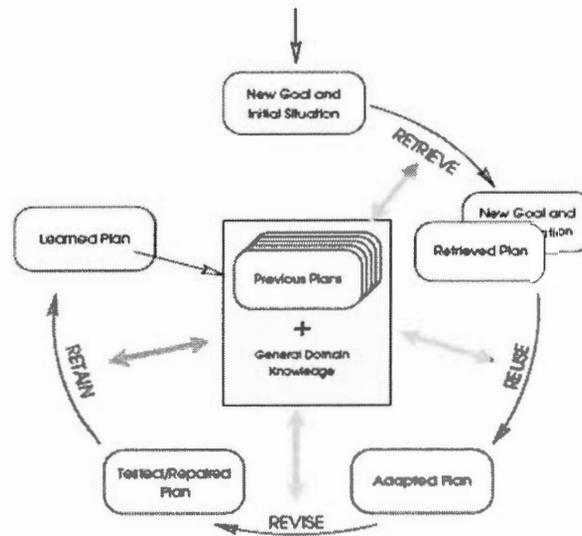
and  $p \in C_s$ ,

Il résout un problème de planification donné si, pour chaque action  $s \neq t_I$  et pour chaque  $p \in C_s$ , il existe un lien causal  $\langle s' \xrightarrow{p} s \rangle$  et pour chaque action  $s \neq t_G$  il existe au moins un lien causal  $\langle s \xrightarrow{q} s'' \rangle$ . Il arrive parfois que durant l'exécution d'un plan, un échec se produise. L'échec peut être causé par une situation inattendue  $F$  ( $F$  étant un ensemble de formules). En CBP, il est intéressant de stocker aussi les échecs pour les éviter par la suite.

En dépit du fait qu'il y ait de multiples représentations de plan mises en œuvre, elles peuvent toutes se rapporter à l'une des catégories suivantes (Spalazzi, 2001) :

- La représentation basée caractéristiques ou relationnelle pure :
- La représentation transformationnelle ou dérivationnelle :
- La représentation indexée ou plate
- La représentation basée logique

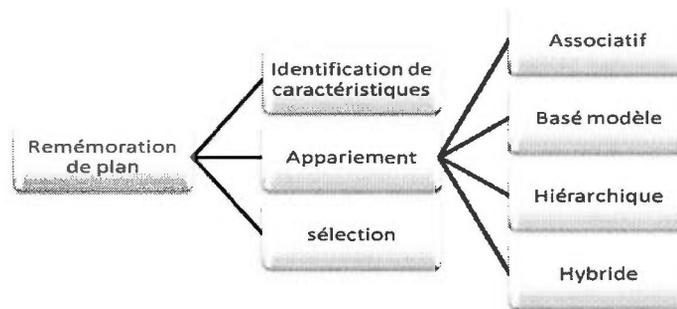
Un CBP peut être considéré comme une instance de CBR spécialisée dans la planification. Son cycle (voir figure 1.6) ainsi que sa décomposition des tâches sont fortement inspirés par ceux proposés dans (Aamodt et Plaza, 1994).



**Figure 1.6** Cycle du CBP (extraite de (Spalazzi, 2001))

La conception d'un planificateur basé cas implique généralement la résolution de problèmes liés aux tâches suivantes (Bergmann et al., 1998; Spalazzi, 2001) :

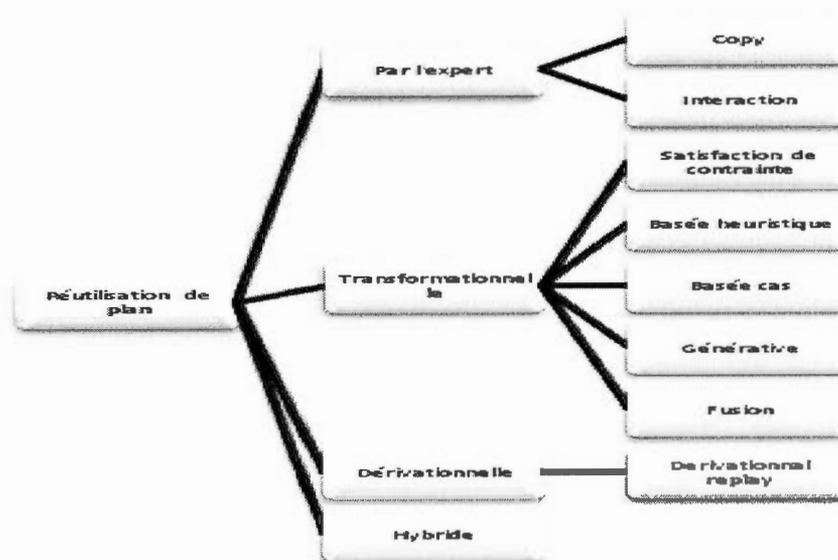
- La remémoration de plan : C'est la récupération en mémoire d'un ou plusieurs plans qui répondent à la requête en cours. Cette tâche est basée sur trois sous-tâches : l'identification de caractéristiques ou dimension à utiliser pour la remémoration (il s'agit généralement du but, de l'état initial et/ou les échecs), l'appariement entre dimensions du nouveau problème à résoudre et ancien plan (de très nombreuses techniques existent, allant d'un appariement de surface à un appariement de concepts plus élaboré) et la sélection qui permettra de choisir le plan à réutiliser.



**Figure 1.7 La tâche de remémoration de plan (adaptée de (Spalazzi, 2001)).**

Le premier niveau (le plus à gauche dans la figure 1.7) décrit la tâche, le deuxième les sous tâches, le troisième les stratégies.

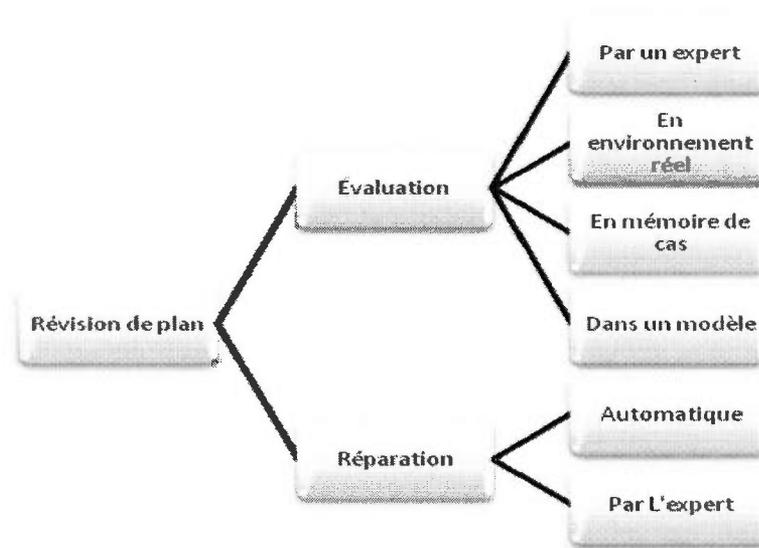
- **La réutilisation du plan :** Il s'agit de la manière de réutiliser et éventuellement adapter un plan récupéré afin de satisfaire le nouveau problème. La nature complexe de la tâche d'adaptation fait que dans bon nombre de systèmes, elle reste du ressort de l'expert du domaine. Lorsqu'elle est faite de manière automatique, de nombreuses familles de techniques existent : l'adaptation transformationnelle qui répare le plan trouvé (elle peut être heuristique, à satisfaction de contrainte, générative, basée fusion ou encore basée cas), l'adaptation dérivationnelle ré-exécute la méthode qui a permis de générer le plan retrouvé en vue de satisfaire le nouveau problème, ou encore une hybridation pour une réutilisation plus efficace.



**Figure 1.8** La tâche de réutilisation de plan (adaptée de (Spalazzi, 2001)).

Le premier niveau (le plus à gauche dans la figure 1.8) décrit la tâche, le deuxième les sous-tâches, le troisième les stratégies.

- La révision du Plan : porte sur l'expérimentation du nouveau plan afin d'évaluer son succès et de le réparer en cas de défaillance. L'évaluation peut être une exécution dans l'environnement réel ou simulé (dépendamment du niveau de risque des tests), une évaluation par l'expert du domaine, une évaluation par l'intermédiaire d'un modèle de simulation ou encore une évaluation dans le cadre de la mémoire de cas (à priori en cherchant des cas similaires à celui généré et en analysant ses échecs ou à posteriori en anticipant les échecs du plan qui sera généré). Une fois le plan évalué, le système peut faire appel à une tâche de réparation manuelle (effectuée par l'expert) ou automatique (utilisant des techniques d'adaptation).



**Figure 1.9** La tâche de révision de plan (adaptée de (Spalazzi, 2001)).

Le premier niveau (le plus à gauche de la figure 1.9) décrit la tâche, le deuxième les sous-tâches, le troisième les stratégies.

- Le stockage du plan : Savoir quand stocker le nouveau plan pour une utilisation future. Habituellement, lorsque le nouveau plan échoue, il est stocké avec la justification de son échec. C'est ce qui représente l'aspect apprentissage du CBP. Cela implique la sélection de la connaissance à retenir, sous quelle forme la retenir et comment réorganiser la mémoire pour la retrouver au moment opportun. Elle se base sur deux sous-tâches, l'extraction et la réorganisation. L'extraction consiste à définir les connaissances du cas qui serviront de source d'apprentissage. Ces connaissances peuvent être tout le cas, les décisions et leurs justifications (en cas d'adaptation dérivationnelle) ou encore les contraintes (en cas d'adaptation basée contraintes). Les échecs et leur explication peuvent aussi être extraits et stockés dans le même cas ou encore dans un cas à part. La réorganisation n'a pas lieu lorsqu'il est possible de déterminer le 'bon' emplacement pour sauvegarder le nouveau cas (ajout à la fin d'une mémoire plate ou dans une arborescence figée). Par contre, elle est

nécessaire lorsque l'organisation est basée sur une réévaluation de métrique de similarité (la pondération des métriques est mise à jour pour accélérer la recherche) ou de l'indexe de recherche. Elle est aussi nécessaire lors d'une modification interactive effectuée par l'expert du domaine.

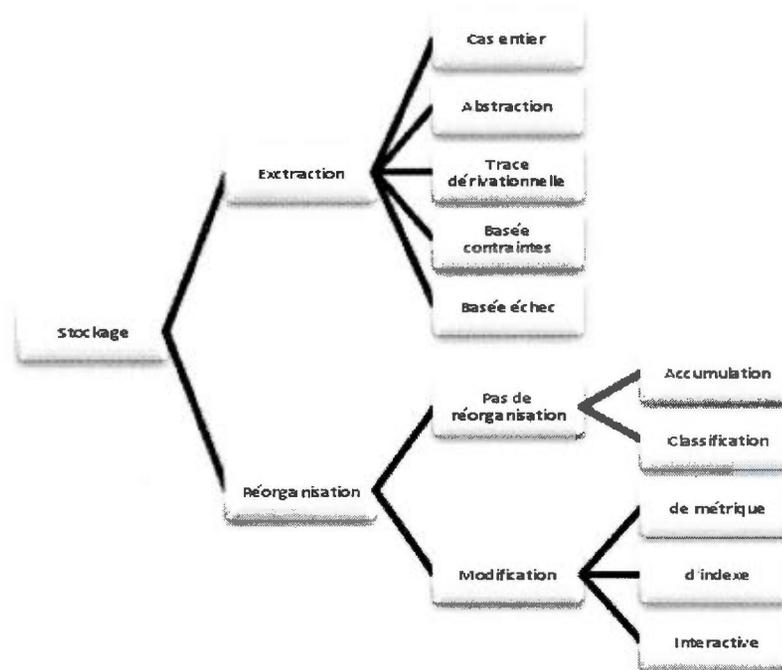


Figure 1.10 La tâche de stockage de plan (adaptée de (Spalazzi, 2001)).

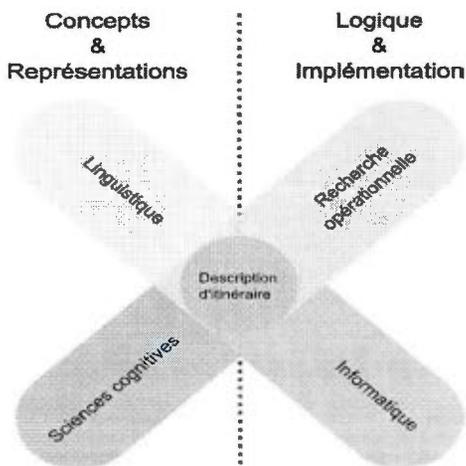
Le premier niveau (le plus à gauche dans la figure 1.10) décrit la tâche, le deuxième les sous tâches, le troisième les stratégies et le dernier (le plus à droite) décrit les méthodes associées à la stratégie.

### 1.8.2 Cadre spécifique à la thèse et caractérisation cognitive du CBP

Dans le cadre de cette thèse, nous nous intéressons à un type particulier de planification qu'est la planification de chemin ou d'itinéraire. D'un point de vue cognitif, la planification d'itinéraire est perçue comme un système dynamique complexe donnant lieu à une interaction intense entre l'environnement qu'est l'infrastructure routière et différents intervenants tels que le conducteur de véhicule,

le contrôleur de trafic, l'intervenant en sécurité, ...etc. Par ailleurs, les services fournis aux conducteurs de véhicules doivent être réfléchis en fonction du profil cognitif de l'utilisateur (perception, préférences, temps de réaction... etc.). À cela s'ajoute une dimension de cognition sociale à travers les interactions entre différents usagers du réseau.

Les sciences cognitives étudient les différents mécanismes de perception mis en œuvre au cours d'une interaction entre l'homme et son environnement (Brosset, 2008). Une description d'itinéraire pouvant être verbale, elle constitue un sujet d'étude pour les chercheurs en linguistique. Leurs études portent essentiellement sur l'analyse des expressions utilisées pour la description d'itinéraires et les mécanismes d'extraction d'informations spatio-temporelles à partir de descriptions en langage naturel. La recherche en informatique, quant à elle, s'intéresse au codage et au décodage des informations logiques, structurelles, et spatio-temporelles contenues dans une description d'itinéraire, la navigation automatisée d'agents artificiels en interaction avec un environnement ou un agent humain. À l'intersection de ces domaines, se trouve la description d'itinéraire (Figure 1.11).



**Figure 1.11** Principaux domaines d'études des descriptions d'itinéraires  
(extraite de (Brosset, 2008))

Les approches traditionnelles de planification utilisent des algorithmes tels que A\* (Farreny et Ghallab, 1987) qui sont guidés par une fonction de coût minimisant la distance. Cependant, en plus du fait que cette approche soit computationnellement prohibitive, il y a une évidence cognitive suggérant que les chemins proposés ne conviennent pas aux besoins et préférences personnelles de l'utilisateur visé. Depuis plusieurs années, cette méthode traditionnelle de planification a été lourdement critiquée par les scientifiques en sciences cognitives à cause de la notion de « one-size-fits-all » qu'elle met en œuvre. McGinty et Smyth, (2001) rapportent qu'il existe de multiples critères pouvant influencer le choix du chemin. Certains de ces critères sont évidents, comme la longueur du chemin, d'autres le sont moins tel que la valeur panoramique d'une route. Le nombre de tournants, la familiarité et la durée du trajet sont des exemples de critères contribuant à la bonne qualité d'une route. Ces critères ne constituent en fait qu'une infime partie des critères influençant le choix d'un chemin. Les auteurs parlent de notion de « different-strokes-for-different-folks » notion fondamentale de planification de chemin dans un contexte personnalisé. Ce qui propulse la planification du chemin dans une dimension d'intelligence ambiante avec ses composants de base, à savoir (Carrillo-Ramos, 2007) : l'ubiquité: (la capacité d'être disponible n'importe où), la sensibilité à la localisation des objets, des appareils et des personnes pour prendre en compte le contexte d'usage, l'interaction naturelle (maximisant l'ergonomie de l'interface homme/machine) et enfin l'intelligence en analysant le contexte et en procédant à l'adaptation dynamique aux situations.

La planification de chemin ou d'itinéraire peut se définir par (Machroum, 2007) :

**Définition notionnelle**

D'après le dictionnaire (Le Robert, 1973), l'itinéraire est "un chemin à suivre pour aller d'un lieu à un autre". La description d'itinéraire est donc un ensemble constitué d'éléments statiques et dynamiques nécessaires pour effectuer ce chemin (description de "chemin à suivre pour aller d'un lieu à un autre"). Le chemin à décrire est situé dans un environnement qui se compose d'objets statiques comme des bâtiments, espaces verts, parkings, chemins piétons, trottoirs, rues, routes, des panneaux d'indications, virages, rond-points, etc. Cependant, quand on parle de "chemin à suivre", on ajoute un

élément dynamique à cet ensemble statique. Plus précisément il s'agit d'un ensemble d'actions visant le déplacement d'un acteur de l'endroit A à l'endroit B.

Du point de vue formel, la description d'itinéraire est une forme d'expression qui contient à la fois des éléments descriptifs et des indications directives. Étant donné son caractère injonctif, elle appartient à la famille de discours procéduraux [...]

**Définition fonctionnelle**

La description d'itinéraire doit permettre à un individu de se déplacer si possible sans erreur dans un environnement inconnu. Afin d'accomplir sa fonction, la description d'itinéraires doit être adaptée aux besoins de l'utilisateur, ce qui veut dire que, d'une part, elle doit tenir compte du type de déplacement, des caractéristiques personnelles de l'utilisateur, de ses motifs du déplacement (loisir, travail), et d'autre part, elle doit être présentée sous le format adéquat à la situation d'utilisation [...]

La description d'itinéraire peut être considérée comme le produit d'une interaction entre un demandeur et son informateur, en réponse à un besoin spécifique du demandeur.

Dans ce cadre spécifique, nous nous intéressons à la planification d'itinéraire personnalisé répondant à des préférences du conducteur. On distingue trois familles de préférences de l'utilisateur (Mokhtari, Pivert et Hadjali, 2009):

- Les préférences spatiales ( $p_s$ ) : l'utilisateur exprime ses préférences concernant les routes, des lieux ou des parties du réseau routier. Elles représentent la plus grande classe de préférence en termes de diversité. Chaque entité du réseau routier peut être concernée par une telle préférence à partir du moment où cette entité a des coordonnées spatiales. Deux catégories de préférences spatiales peuvent être identifiées:
  - Les préférences sur un ou plusieurs éléments spécifiques d'une route du réseau. Elles correspondent aux préférences sur un ou plusieurs sommets/arêtes d'un réseau routier, à savoir, l'intersection. Cette catégorie est utilisée par un conducteur en faveur ou défaveur de certains éléments spécifiques du réseau. Un exemple est «d'éviter les autoroutes à péage ».

- Les préférences sur une partie induite par un réseau routier (sous-graphe induit). Elles correspondent aux préférences qui concernent une partie induite d'un réseau routier, appelée zone. Une zone est une zone géographique, par exemple une zone administrative. Les préférences sur une zone sont sur l'environnement qui entoure un itinéraire. Un exemple est «préférer une route qui passe à travers la zone administrative R ».
- Les préférences spatio-temporelles ( $p_{st}$ ): ce sont des préférences spatiales comportant un élément de temps qui exprime le moment ou la période durant laquelle la préférence spatiale est pertinente. Un exemple en est "éviter le centre-ville aux environs de midi". Ce sont des préférences complexes. Elles peuvent comporter différents types d'entités et de relations temporelles. Trois types d'états temporels peuvent être impliqués dans une planification de la requête:
- les déclarations se référant à un point précis dans le temps, appelé instant, désigné par  $i$ . En représentation mathématique classique un instant  $i$  est considéré comme un singleton représentant une date / heure (par exemple: 20 heures).
  - les déclarations qui font référence à un temps continu période dont la durée est appelé intervalle, notée  $I$  et mathématiquement représenté par une limite inférieure et supérieure liées (instant début, instant fin)  $[-i, +i]$ .
  - les déclarations qui font référence à une durée (par exemple, la distance absolue entre deux instants), notée  $d$  représentée par un début et une durée  $[-i, -i+ d]$ .
- Les préférences intrinsèques ( $p_{int}$ ) elles concernent des propriétés globales d'une route. Contrairement aux préférences spatiales et spatio-temporelles qui sont locales dans la mesure où elles concernent une partie d'un itinéraire, les préférences

intrinsèques caractérisent la route dans son ensemble. Ils comportent des critères qualitatifs, tels plus cher, rapide, sûr... Parmi les propriétés plus intuitives d'un itinéraire, nous pouvons citer: la rapidité, la longueur, la sécurité, coût et le confort. D'autres propriétés peuvent être d'intérêt dans certains contextes spécifiques, par exemple: la robustesse (pour des applications militaires ou les services de secours) ou de la pollution produite.

La modélisation de ces connaissances ainsi que l'inférence s'y attachant prendront une orientation différente.

La planification présente donc plusieurs aspects importants identifiés par Reignier dans (1992), à savoir:

- La résolution automatique de problèmes: la solution est l'assemblage d'un ensemble d'opérateurs de base (ou actions élémentaires) élaborée lors de la planification pour permettre d'atteindre un objectif fixé.
- La prévision : un plan est une prévision d'une séquence d'actes à accomplir (et de l'évolution consécutive de l'environnement).
- La création d'une nouvelle connaissance: un plan nouveau accroît la base de connaissances du système: le plan, une fois élaboré, peut être conservé sous une forme généralisée et être réutilisé lorsqu'un problème similaire se posera (généralisation de plans, domaine de l'apprentissage). De même, la déduction qui peut être réalisée sur l'information rend explicites des connaissances qui jusque-là étaient implicites (par exemple, déduction de faits à partir de théorèmes).

Koehler (1994) quant à elle a pu identifier des pièges possibles pour la planification de second principe :

- Lorsqu'appliquées à des domaines où la planification de premier principe a été performante, les techniques de second principe ne conduisent pas toujours à un accroissement de l'efficacité. Des recherches supplémentaires sont nécessaires

pour identifier le domaine d'application pour la planification basée cas contrairement à la planification générative.

- Des tâches complexes de planification peuvent nécessiter la production des plans contenant les structures de contrôle comme les boucles ou la récursivité. Le réaménagement de ces plans est extrêmement difficile et doit être étudié plus en avant. En général, la réutilisation des plans complexes est très utile lorsque le plan peut être instancié pour la solution souhaitée et aucune opération coûteuse d'adaptation ne doit être effectuée.

Même si l'étude est relativement ancienne, ces conseils restent de rigueur.

## 1.9 Conclusion

Il est certain que la richesse du paradigme CBR n'a pu être dépeinte avec exactitude à travers ce bref chapitre, l'objectif étant de fournir les concepts de base nécessaires pour aborder les chapitres suivants. Néanmoins, les références citées (Ashley et Rissland, 1988 ; Rissland et Skalak, 1989; Kolodner, 1991 ; Kolodner, 1992 ; Kolodner, 1993 ; Aamodt et Plaza, 1994 ; Watson, 1997 ; Lopez de Mantara et al, 2005) peuvent fournir le complément d'information nécessaire.

Par ailleurs, la présentation des concepts inhérents au CBP s'est faite selon une vision de modélisation sans aborder des illustrations spécifiques pour deux raisons : nous n'avons présenté que les outils de modélisation qui nous facilite le travail à faire au chapitre IV. Nous avons évité de surcharger la présentation par des exemples de planificateurs basés cas puisque l'objectif n'étant pas le planificateur en soi, mais le modèle de mémoire qui va le soutenir. Or dans la littérature CBP, l'emphase n'est pas mise sur cet aspect. Cela aurait été donc plus déroutant que bénéfique.

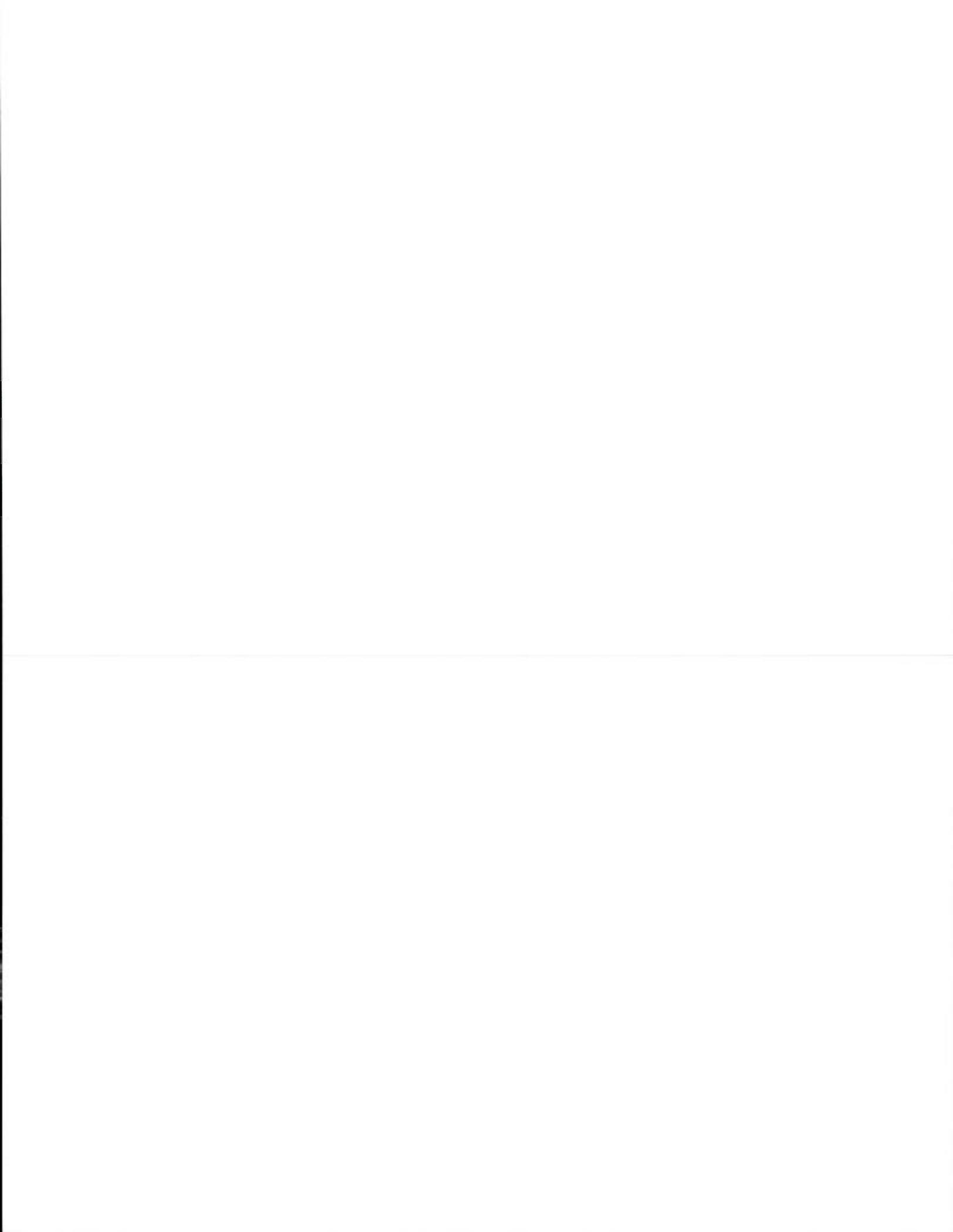
Ce qu'il y a lieu de retenir est que l'adaptation de plan dans le cadre de la génération de plan de second principe peut être coûteuse si le plan trouvé ne s'y prête

pas où n'est pas trouvé dans un délai raisonnable. Nous proposerons dans la suite de ce travail une approche de minimisation du coût en misant sur la recherche spécifique de plans adaptables. Auparavant, nous présentons dans le chapitre qui suit l'outil de base de la recherche sur lequel nous grefferons le critère de réutilisabilité. Il s'agit de l'approche PSO qui est à la base un outil d'optimisation, mais que nous avons adaptée à nos fins.

## CHAPITRE II

# L'optimisation par essaim de particules

1. Introduction.
2. La genèse de la Swarm Intelligence.
3. L'inspiration éthologique.
4. Le modèle de l'essaim.
5. La PSO de base.
6. Variantes et hybridations de la PSO.
7. Situation de la PSO.
8. Conclusion.



## 2.1 Introduction

La Swarm Intelligence (SI) ou intelligence en essaim fait référence à un paradigme de calcul distribué basé sur l'intelligence collective qui émerge de la coopération de multiples agents autonomes (Norman, 2006). La SI applique des concepts basés sur le fourrage des colonies de fourmis et le regroupement d'animaux sociaux tels les oiseaux au calcul basé agents. Les trois paradigmes majeurs de la SI sont : l'optimisation par colonies de fourmis (« Ant Colony Optimization » ou ACO), l'optimisation par essaims de particules (« Particle Swarm Optimization » ou PSO) et les essaims robotiques (« Swarm Robotics » ou SR).

Nous nous intéressons dans ce chapitre à mieux comprendre ces modèles d'inspiration éthologique<sup>7</sup>, à en décortiquer le comportement (contrôle et émergence), et le mode de communication (interaction et stigmergie).

## 2.2 La genèse de la Swarm Intelligence

La Swarm Intelligence (SI) est une technique de l'intelligence artificielle basée sur l'étude du comportement collectif de manière décentralisée et autoorganisée. L'expression "Swarm Intelligence" a été présentée par Beni et Wang (1989), dans le contexte des systèmes robotiques cellulaires.

Dans (Beni, 2005b), Gerardo Beni rapporte comment il a utilisé pour la première fois le terme «essaim» pour ses robots cellulaires. Il retrace la genèse de termes tels que «essaim» et «intelligence en essaim», d'une manière très détaillée avec une argumentation scientifique rigoureuse. Dans ce qui suit, nous reproduisons la matière essentielle de ce document. Nous avons pu noter que la principale réflexion a été faite en termes robotiques, mais demeure pertinente dans le cadre général.

---

<sup>7</sup> Au sens étude des mœurs et des comportements spontanés des animaux sociaux dans leur environnement naturel.

Il y a trente ans de cela, il y avait un engouement important pour les automates cellulaires. Beni s'y était intéressé, car ceux-ci pouvaient produire des modèles de complexité significative à partir de règles simples. À la même époque, au Japon, Fukuda avait utilisé le terme "robots cellulaires" indiquant les groupes de robots qui peuvent fonctionner comme les cellules d'un organisme et qui s'assemblent en structures plus complexes. Beni avait également utilisé le terme "robot cellulaire" mais pour indiquer une extension, ou un type plus général des automates cellulaires. L'extension résidait dans le fait que les unités de l'automate n'opéraient ni de manière synchrone, ni de façon séquentielle et se déplaçaient et interagissaient dynamiquement. Ils étaient censés représenter un groupe de robots autoorganisés simples. Beni a utilisé le mot "Swarm" qu'Alex Meystel avait mentionné lors d'une discussion. Il y avait pour cela quelques bonnes raisons. Le fait est que le groupe de robots de Beni et Wang n'était pas seulement un "groupe". Il avait quelques caractéristiques particulières, qui se trouvaient dans les essaims d'insectes, c'est-à-dire, le contrôle décentralisé, le manque de synchronisme, des membres simples et (quasi) identiques. De plus, la taille, c'est-à-dire, le nombre d'unités, avait son importance. Elle n'était pas assez grande pour être traitée avec des moyennes statistiques, pas assez petite pour être traitées comme des problèmes de complexité réduite. Le nombre d'unités a été pensé pour être réaliste de l'ordre de  $10^2$  à  $10^{<<23}$ . Ainsi "Swarm" était le terme tout à fait approprié pour distinguer ce type de groupe. Les essaims sont plus attrayants pour les systèmes robotiques, par rapport aux systèmes centralisés conçus pour la même tâche, ils ont des composants très simples. Ainsi, les unités de robots pourraient être modulaires, produites en masse, et pourraient être interchangeables et disponibles. Le second avantage est la fiabilité: puisque l'essaim est en général hautement redondant, l'essaim pourrait être conçu de manière à survivre même à de nombreux types de perturbations (pouvant être plus sévères que celles considérées dans la norme des systèmes de contrôle). En raison de la redondance, l'essaim aurait la capacité de s'adapter dynamiquement à l'environnement de travail, ce qui constitue un autre élément nécessaire pour une

grande fiabilité. Il est aussi possible d'envisager l'essaim agissant comme un solide système de calcul parallèle et, par conséquent, effectuer des tâches au-delà de celles possibles à d'autres types de systèmes robotiques, qu'ils soient complexes uniques centralisés de robots ou de groupes de robots.

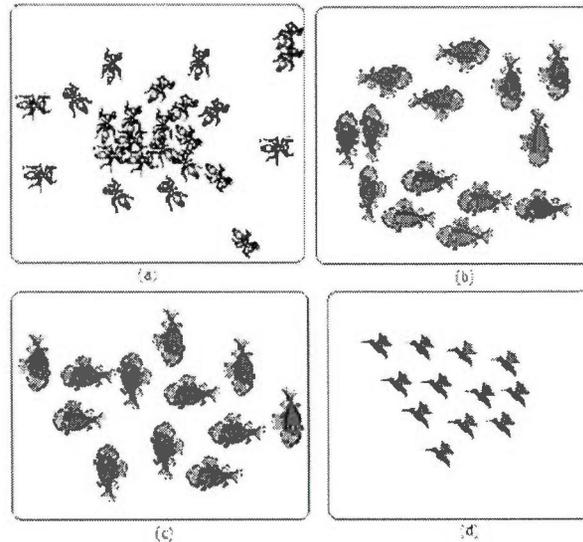
Il n'est pas surprenant que la notion d'essaim apparaisse de plus en plus étroitement associée avec des systèmes capables de mener à bien non seulement des tâches utiles, mais également des tâches "intelligentes". D'un point de vue robotique, les essaims s'autoorganisent en modèles. Du côté biologique, ils construisent des modèles ordonnés ce qui constitue une caractéristique de l'intelligence. Par ailleurs, la reconnaissance et/ou l'analyse des schémas, que réalisent les essaims pour optimiser une fonction sont aussi des tâches intelligentes. Ainsi, de tous les points de vue, nous sommes amenés à voir les essaims comme une chose intelligente, d'où le terme — "Swarm Intelligence".

### 2.3 L'inspiration éthologique.

Alors que les roboticiens essayaient de faire faire à l'essaim certaines tâches prescrites, les biologistes ont essayé d'expliquer le comportement des insectes comme les sociétés d'essaims. Quatre comportements dynamiques collectifs sont illustrés à la figure 2.1. (Abraham et Grosan, 2006):

- Essaim: un agrégat avec cohésion, mais un faible niveau de polarisation (alignement parallèle) entre les membres.
- Torus: les individus en perpétuelle rotation autour d'un vide de base (fraisage). Le sens de rotation est aléatoire.
- Groupes parallèles dynamiques: les individus sont polarisés et se déplacent comme un groupe cohérent, mais des individus peuvent circuler en dehors du groupe et la densité et la forme du groupe peuvent fluctuer.
- Groupes hautement parallèles: beaucoup plus statique en termes de l'échange de positions spatiales au sein du groupe que la dynamique de

groupe parallèle et la variation de la densité et la forme est minime.



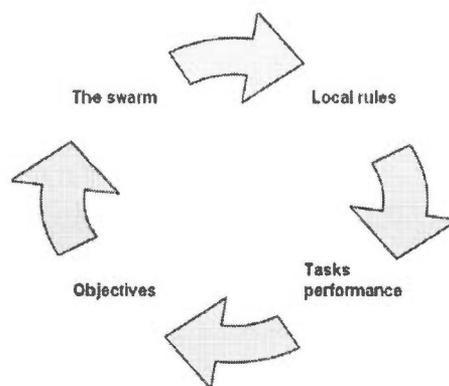
**Figure 2.1** Modèles de comportements collectifs: (a) essaim (b) torus (c) groupe parallèle dynamique et (d) groupe hautement parallèle (extraite de (Abraham et Grosan, 2006)).

Une étude plus détaillée a été publiée dans (Garnier, Gautrais et Theraulaz, 2007). Les auteurs examinent les principes biologiques qui sous-tendent l'organisation des colonies d'insectes. Ils commencent avec quelques rappels sur le caractère décentralisé de ces systèmes et ensuite décrivent les mécanismes sous-jacents aux comportements collectifs complexes des insectes sociaux, de la notion de stigmergie à la théorie de l'autoorganisation dans les systèmes biologiques. Ils soulignent en particulier le rôle des interactions et l'importance des bifurcations qui apparaissent dans la production collective de la colonie lorsque certains des paramètres du système changent. Ils abordent le rôle des modulations de comportements individuels par les perturbations (soit l'environnement ou à l'intérieur de la colonie) dans la flexibilité générale de colonies d'insectes. Ils concluent que les futures études sur les comportements biologiques autoorganisés devraient investiguer sur ces modulations pour mieux comprendre comment les colonies d'insectes s'adaptent aux mondes

incertains. Nous verrons dans la section 5 du présent chapitre, la façon avec laquelle ces modulations ont été mises en équation dans le cadre de la PSO.

Les qualités principales d'une société d'animaux sont: la décentralisation, le non-synchronisme, avec quasi-homogénéité, la simplicité, pas en grand nombre. Un concept clé est la "Stigmergie", c'est-à-dire, la communication par le biais de l'environnement (Kennedy et Eberhart, 2001; Engelbrecht, 2007; Bonabeau, Dorigo et Theraulaz, 1999). Les fourmis communiquent à d'autres fourmis la «qualité» d'un chemin, par un marquage avec des phéromones de sorte qu'un mécanisme de rétroaction positive se termine finalement quand la plupart des insectes suivent le « meilleur » chemin. C'est un exemple d'«optimisation », et dans ce domaine particulier, la notion d'essaim avait plus de succès.

Pour en revenir au concept d'essaim, à un haut niveau, celui-ci peut être considéré comme un groupe d'agents coopérant pour atteindre un comportement déterminé et atteindre un certain objectif (voir Figure 2.2). Cette intelligence collective semble se dégager de grands groupes de simples agents. Les agents utilisent des règles simples pour régir leurs actions et par les interactions de l'ensemble du groupe, l'essaim atteint ses objectifs. Un type d'autoorganisation se dégage de la collection des actions du groupe.

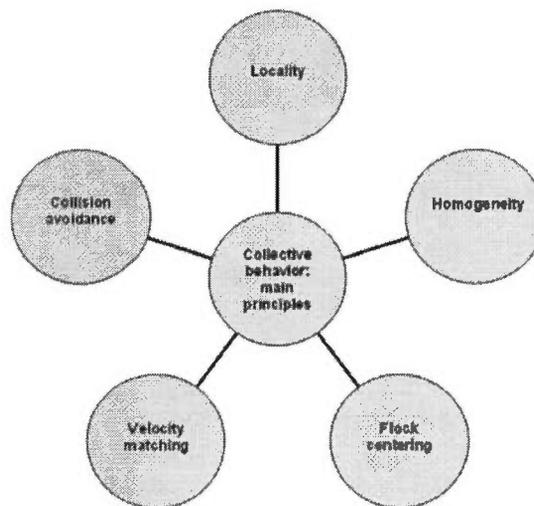


**Figure 2.2** Le simple schéma d'action de l'essaim (extraite de (Abraham et Grosan, 2006)).

Un agent autonome est un sous-système qui interagit avec son environnement, qui consiste probablement en d'autres agents, mais agit de manière relativement indépendante de tous les autres agents (Abraham et Grosan, 2006). L'agent autonome ne suit pas les ordres d'un dirigeant. Par exemple un oiseau règle ses mouvements pour assurer la coordination avec les mouvements de son groupe, généralement ses voisins proches. Un oiseau dans une volée tente simplement de rester proche de ses voisins, mais éviter les collisions avec eux. Chaque oiseau ne prend pas les commandes à partir de n'importe quel chef d'oiseaux, car il n'existe pas de leader des oiseaux. Le comportement en essaim aide l'oiseau à tirer avantage de plusieurs choses, notamment la protection contre les prédateurs (en particulier pour les oiseaux au milieu de la volée), et la recherche de nourriture (chaque oiseau exploite les yeux de tous les autres oiseaux).

De nombreux groupes d'animaux tels que les poissons et les oiseaux affichent clairement l'ordre structurel, avec le comportement des organismes de manière très intégrée, même s'ils changeaient de forme et de la direction, ils semblent se déplacer comme une entité unique et cohérente. Les grands principes du comportement collectif de l'essaim tels que présentés en figure II.3, sont les suivants:

- Homogénéité: tous les oiseaux dans le groupe ont le même modèle de comportement. Le groupe se déplace sans un leader, même si des dirigeants temporaires semblent apparaître.
- Localité: le mouvement de chaque oiseau est seulement influencé par ses plus proches compagnons. La Vision est considérée comme le sens le plus important pour l'organisation.
- Prévention des collisions: éviter trop de proximité.
- L'appariement de vitesse: essayer de trouver des correspondances avec la vitesse du voisinage.
- Centrage du groupe: tenter de rester proche du voisinage.



**Figure 2.3** Principes du comportement collectif de l'essaim (extraite de (Abraham et Grosan, 2006)).

Les individus tentent de maintenir une distance minimale entre eux et les autres en tout temps. Cette règle a la priorité la plus élevée et correspond à un comportement souvent observé chez les animaux dans la nature. Si les individus n'effectuent pas une manœuvre d'évitement, ils ont tendance à être attirés vers d'autres individus (pour éviter d'être isolé) et de s'aligner avec les voisins.

#### 2.4 Modèle de l'essaim

Comme dans toute science cognitive, la recherche en intelligence par essaim englobe à la fois la science et l'ingénierie. La science est axée sur la nécessité de comprendre et de modéliser naturellement les systèmes à essaims. Plus récemment, un nombre croissant de chercheurs issus du domaine de l'ingénierie ont étudié la manière de créer et de contrôler des essaims matériels ou logiciels. Cette forme d'ingénierie souvent s'inspire d'études scientifiques de l'essaim naturel. L'intérêt scientifique porté aux essaims naturels nécessite une justification. Il convient

toutefois de se poser la question: pourquoi sommes-nous intéressés à la conception d'objets essaims (qu'ils soient logiciels ou matériels) et pourquoi essayons-nous de résoudre des problèmes par leur biais? Qu'est-ce que ces systèmes fourniront que les technologies actuelles ne peuvent pas?

Les créateurs d'essaims artificiels tentent de copier les succès observables et l'efficacité des systèmes d'essaims naturels. Malgré la relative simplicité de ses agents, les systèmes naturels d'essaims peuvent souvent faire preuve d'un comportement extrêmement complexe. Les systèmes à essaims naturels sont hautement évolutifs, ils sont parfois composés de plusieurs millions d'individus. En outre, ces systèmes ont tendance à être souples et robustes. Ils réagissent bien à l'évolution rapide des environnements, et continuent à fonctionner même si bon nombre de leurs individus sont rendus incapables. Des études ont montré que, dans de nombreux cas, de simples règles de comportement au niveau de l'individu sont suffisantes pour expliquer le comportement complexe. Ces modèles ne nécessitent pas de communication globale. Ils s'appuient uniquement sur des dispositifs locaux de détection et de communication.

Les chercheurs ont commencé à utiliser des modèles de comportement dans des essaims artificiellement créés. Dans le domaine de la robotique, des systèmes robotiques en essaims sont attrayants, puisque les robots peuvent être simples, fiables et bon marché. Grâce à la coopération, ces robots relativement simples peuvent ainsi accomplir des tâches difficiles de manière robuste, évolutive et émergente. Des résultats très encourageants ont été obtenus, en particulier dans des applications d'optimisation, optimisation par colonie de fourmis, l'optimisation par essaim de particules, où des essaims d'agents logiciels coopèrent à la recherche de bonnes solutions aux problèmes d'optimisation difficiles. La Swarm Intelligence (SI) fournit ainsi un nouveau cadre pour la conception et la mise en œuvre des systèmes basés sur

plusieurs agents qui sont capables de coopération pour la résolution de problèmes complexes. Les avantages potentiels de l'approche SI sont multiples (Dorigo, 2007):

- Robustesse collective: l'échec de composantes individuelles ne peut pas gêner les performances globales;
- Simplicité individuelle: le comportement coopératif rend possible la réduction de la complexité des individus;
- Évolutivité: les mécanismes de contrôle utilisés ne sont pas fonction du nombre d'agents dans l'essaim.

D'un point de vue ingénierie, ces trois caractéristiques seraient à elles seules des critères de pertinence pour en faire une nouvelle technologie de la connaissance.

La pertinence sociale est patente dans la mesure où les origines de l'intelligence par essaim sont profondément ancrées dans l'étude éthologique des comportements autoorganisés chez les insectes sociaux.

Durant les années 40 et 50, une autre époque des sciences cognitives vit le jour : l'étude des propriétés émergentes des systèmes. L'émergence est une idée importante. Cela signifie qu'après un point de seuil, un système acquiert subitement une propriété qui n'avait pu être prédite à partir de la somme de ses parties. Les essaims constituent un exemple parfait de comportement émergent.

C'est dans le domaine de la philosophie que le thème de l'autoorganisation trouve son origine. Ce mécanisme est parfaitement illustré par les essaims.

Dans un modèle d'essaim, les unités fonctionnent sans contrôle central et aucune horloge globale. Plus précisément, ils fonctionnent de manière partiellement synchrone, ils ne fonctionnent pas simultanément, mais ils ne fonctionnent pas non plus de manière strictement asynchrone, comme, par exemple, dans la mise à jour séquentielle. Les unités d'un essaim exécutent généralement de manière partiellement

synchrone, un algorithme distribué. Dans de nombreux cas, l'algorithme consiste à résoudre un système d'équations linéaires de différence, chaque unité autonome met à jour son état basé sur une équation linéaire avec des valeurs obtenues à partir de la voisine. Habituellement, les unités sont identiques, mais ce n'est pas une restriction nécessaire. Beni (2005a) généralise la définition d'essaim pour inclure une plus large classe de systèmes physiques, et sans restriction à une dimension. Sans formalisme mathématique, il définit un essaim comme suit:

*Définition 1.* Un essaim est un ensemble ordonné de  $N$  unités décrites par  $N$  composantes d'un vecteur, chacune des  $N$  unités peut mettre à jour le vecteur, à son propre temps, en utilisant une fonction des composantes du vecteur.

Dans de nombreux cas pratiques, la fonction est linéaire (essaim linéaire) et la mise à jour est également linéaire (essaim strictement linéaire). Aussi, dans la plupart des cas, chaque unité de l'essaim utilise la même règle de mise à jour avec le même taux moyen de mise à jour (essaim uniforme). La non-uniformité peut concerner la règle ou la moyenne des mises à jour. La règle de mise à jour peut être déterministe ou probabiliste (essaims stochastiques). Par ailleurs, les essaims particulièrement intéressants sont ceux dont la mise à jour de chaque composante se fait à partir des composantes des plus proches voisins ou voisins des plus proches voisins (essaims à faible ordre). Certains essaims ont des unités qui mettent à jour d'autres unités (les essaims gouvernants), mais de nombreux essaims sont autonomes, c'est-à-dire, chaque unité se met à jour automatiquement. Enfin, il convient de noter le cas où une fonction de la description du vecteur est conservée dans la mise à jour (essaims conservateurs).

Il existe deux caractéristiques communes aux paradigmes SI qui mènent à l'intelligence émergente. Le premier étant l'utilisation de boucles de retour, le second est l'utilisation de la communication stigmergique. Celle-ci fait référence à une forme de communication indirecte via l'interaction avec l'environnement.

Les trois paradigmes majeurs de la SI sont : l'optimisation par colonies de fourmis (« Ant Colony Optimization » ou ACO), l'optimisation par essaims de particules (« Particle Swarm Optimization » ou PSO) et les essaims robotiques (« Swarm Robotics » ou SR).

### *La Swarm Robotique*

Notons que la Swarm Robotique (SR) est une discipline appliquant les principes de la SI à des implémentations matérielles (Sahin, 2005). La SR s'inspire de la nature pour implémenter les schémas de coordination multirobots.

Comme tout nouveau terme, le terme "Swarm Robotique" courait le risque de se transformer en 'Buzzword' qui aurait été attaché à des approches peu ou pas en rapport avec l'esprit de la chose. Ces abus auraient, avec le temps, dévié le terme de son sens originel. Pour empêcher cela, Erol Sahin, dans (Sahin, 2005), propose une définition:

**Définition2.** La Swarm Robotique est l'étude de la façon avec laquelle un grand nombre d'agents relativement simple et physiquement incarnés peuvent être conçus de sorte qu'un comportement collectif souhaité émerge à partir des interactions locales entre les agents et entre les agents et l'environnement.

Cette définition est complétée par un ensemble de critères de distinction pour l'approche SR. Les individus doivent avoir une incarnation physique dans le monde et peuvent physiquement interagir avec le monde et être autonomes. Les réseaux de capteurs qui se composent d'éléments de détection distribués, mais sans capacité physique d'action ne devraient pas être considérés comme un essaim de robots.

Les systèmes robotiques métamorphiques, pour lesquels les unités adhèrent les unes aux autres et ne peuvent se déplacer que les unes sur les autres par la formation et la déconnexion des liens avec d'autres unités peuvent aussi être considérées comme

des systèmes robotiques à essaim tant qu'il n'existe pas de planification centralisée et de centre de contrôle. Les études faites sur un petit nombre de robots et ne visant pas l'évolutivité, n'entrent pas dans la robotique en essaim. Bien que mettre une limite inférieure de la taille du groupe soit difficile à justifier, il est admis une taille des groupes de 10 à 20 comme "essaim." (Sahin, 2005). Malgré la baisse des coûts des robots, l'entretien et l'expérimentation avec de grands groupes de robots restent tout de même un obstacle. Par conséquent, la question de la pertinence impose que le terrain reste être ouvert à des études qui sont réalisées avec des tailles de groupes plus petits, mais avec la vision de l'évolutivité en vue.

### *L'ACO*

L'ACO fut introduite dans les années 1990 (voir (Dorigo, Maniezzo et Colomi, 1996; Dorigo, Birattari et Stützle, 2006; Dorigo et Socha, 2007; Gutjahr, 2007)). C'est une métaheuristique inspirée du comportement coopératif des colonies de fourmis. Ces dernières accomplissent des tâches impossibles à accomplir par une fourmi seule. Ceci est réalisé grâce à la coopération et la communication entre plusieurs fourmis constituant une unité fonctionnelle autoorganisée et qui tente d'accomplir la même tâche.

L'ACO est basée sur le fourrage de nourriture des fourmis (Norman, 2006). Les fourmis cherchent le plus court chemin du nid vers la source de nourriture. Durant le fourrage, les fourmis secrètent des phéromones leur permettant de retrouver leurs chemins vers le nid et éventuellement vers la source. La phéromone secrétée constitue une piste qui sera suivie par d'autres fourmis durant leur fourrage. Plus nombreuses sont les fourmis suivant cette piste, plus forte est la piste. La première fourmi retournant de la source aura une trace de phéromone plus puissante que celles qui ne sont pas revenues encore puisqu'elle est passée par le même chemin deux fois. Ainsi les autres fourmis seront plus tentées par traverser le même chemin qui deviendra la piste la plus prépondérante graduellement. Inversement, sur les chemins de moins en

moins utilisés, il y a évaporation de la phéromone. Le chemin résultant sera le chemin le plus court.

L'utilisation de la phéromone est un exemple de communication stigmergique puisque les fourmis ne communiquent pas de manière directe, mais altèrent leur environnement et communiquent à travers.

Cette technique de fourrage exploite le feedback positif et l'évaluation implicite de la solution qui sont deux aspects clés de la Swarm Intelligence. La solution est implicitement évaluée dans la mesure où le plus court chemin est estimé très tôt. Parce qu'il est graduellement renforcé (plus elle est renforcée plus la piste est suivie et plus elle est suivie plus elle sera renforcée), on parlera de feedback positif.

L'ACO a été appliquée avec succès à divers problèmes d'optimisation (Norman, 2006), qu'il s'agisse de problèmes statiques (dont les caractéristiques ne changent pas durant toute la vie du problème, tels le voyageur de commerce ou le coloriage de graphe) ou dynamiques (qui démarrent d'un état initial puis évoluent, tel le routage dans un réseau téléphonique).

### ***La PSO***

Les oiseaux se déplacent comme un groupe cohésif sans un chef désigné. Le groupe a tendance à converger vers la source de nourriture ou encore les positions perchoirs. La PSO s'inspire de ce comportement afin d'approximer des solutions pour des problèmes.

Le comportement des groupes d'oiseaux ou des bancs de poissons est une propriété émergente d'une règle d'interaction entre individus (Norman, 2006; Clerc, 2005; Kennedy et Eberhart, 2001). Cette règle consiste au fait que chaque individu du groupe cherche à rester avec l'autre (on parle d'attraction), mais en même temps, il garde une distance de sécurité vis-à-vis de son voisinage pour éviter les collisions (on

parle de répulsion). C'est cet équilibre entre l'attraction et la répulsion qui fait que le groupe soit cohésif.

Les individus sont aussi attirés par les sources de nourriture ou les perchoirs, ce qui les pousse à quitter le groupe, ce qui pousse par là même, les individus voisins à les suivre et à travers la boucle de retour positive le reste du groupe suivra la ressource.

Grâce à un équilibre subtil entre l'attraction globale du groupe, la répulsion locale du voisinage et l'attraction locale des ressources, le rassemblement devient un comportement efficace pour le fourrage.

Les individus utilisent la communication stigmergique de manière subtile. Leur position dans l'environnement influence le comportement de leurs proches voisins. Ils communiquent avec le groupe simplement en changeant de position.

La PSO applique ce modèle de comportement pour contrôler les actions des agents simples (particules) qui se déplacent dans un espace d'état associé à un problème et cherche à converger vers la solution optimale. L'espace problème est de dimension  $n$  et est caractérisé par une fonction à optimiser. Au début de l'exécution de l'algorithme, on affecte aux particules une fonction objective (« fitness »). Tout au long de l'exécution, les particules se déplacent dans l'espace d'état et évaluent constamment la valeur de leur position courante. Si cette valeur respecte un certain seuil, la particule s'établit alors à ce point. Comme l'algorithme progresse, les particules vont converger vers une position particulière qui correspond à une solution du problème. Le terme s'établir correspond ici au fait que lorsque la « fitness » d'une position pour une particule augmente sa vitesse diminue. Les oiseaux utilisent une heuristique stochastique définie par un algorithme qui indique leur désir de chasser relativement à leur désir de s'établir sur un perchoir. Cette heuristique est utilisée

lorsqu'on doit déterminer si une particule doit s'établir sur une solution ou continuer à chercher pour une solution meilleure.

## 2.5 La PSO de base

Dans un système PSO, un essaim d'individus parcourt l'espace de recherche. Chaque particule représente une solution candidate au problème d'optimisation. La position d'une particule est influencée par la meilleure position visitée par elle-même (c'est-à-dire sa propre expérience) et la position de la meilleure particule de son voisinage (c'est-à-dire l'expérience des particules voisines). Lorsque le voisinage d'une particule est l'essaim en entier, la meilleure position dans le voisinage est considérée comme la meilleure particule, et l'algorithme est appelé le *gbest* PSO. Quand des voisinages plus petits sont utilisés, l'algorithme est généralement considéré comme le *lbest* PSO. La performance de chaque particule (c'est-à-dire sa proximité de l'optimum global) est mesurée au moyen d'une fonction objective  $\psi$  qui varie avec le problème d'optimisation.

La PSO est fondée sur un essaim de  $M$  individus appelés particules, chacune représentant une solution à un problème à  $N$  dimensions. Son génotype est composé de  $2 \times N$ , les  $N$  premiers représentant les coordonnées des particules dans l'espace de recherche, et les  $N$  derniers, les composantes de leur vitesse. Du point de vue de l'évolution, une particule se déplace avec une vitesse adaptable au sein de l'espace de recherche et conserve dans sa propre mémoire la meilleure position jamais atteinte. Les paramètres sont modifiés lors du passage d'une itération à l'autre comme décrit ci-dessous. La vitesse  $v_{i,d}(t + 1)$  de la  $i$ -ème particule ( $d$  indiquant la dimension de l'espace,  $d=1, \dots, N$ ) à la prochaine étape  $t+1$  est une combinaison linéaire de la vitesse courante  $v_{i,d}(t)$  de la  $i$ -ème particule à l'instant  $t$ , de la différence entre la position  $b_{i,d}(t)$  de la meilleure solution trouvée jusqu'à ce jour (ramenée à la dimension  $d$ ) et de  $p_{i,d}$  la position actuelle de la  $i$ -ème particule (ramenée à la dimension  $d$ ), et de la différence entre la meilleure position jamais trouvée dans la

population  $b_{g,d}(t)$  (dans la dimension  $d$ ) et celle de la  $i$ -ème particule  $p_{i,d}(t)$  (dans la dimension  $d$ ):

$$v_{i,d}(t+1) = w \cdot v_{i,d}(t) + c_1 \cdot u(0,1) \otimes (b_{i,d}(t) - p_{i,d}(t)) + c_2 \cdot u(0,1) \otimes (b_{g,d}(t) - p_{i,d}(t)) \quad (1)$$

$$\text{pour } i=1, \dots, M \quad \text{et} \quad d=1, \dots, N$$

où  $\otimes$  désigne la multiplication des vecteurs point par point,  $u(0,1)$  est une fonction qui retourne un vecteur dont les positions sont générés au hasard par une distribution uniforme dans  $[0, 1]$ ,  $c_1$  est le paramètre cognitif,  $c_2$  est le paramètre social, et  $w$  est le facteur d'inertie dont la portée est de  $[0.0, 1.0]$ . Les valeurs de vélocité doivent être dans une fourchette définie par deux paramètres  $v_{min}$  et  $v_{max}$ . Une amélioration de la PSO originelle est de faire varier  $w$  au cours de l'exécution, à partir d'une valeur maximale  $w_{max}$ , de le décrémenter linéairement à mesure que le nombre d'itérations augmente jusqu'à une valeur minimale  $w_{min}$  comme suit:

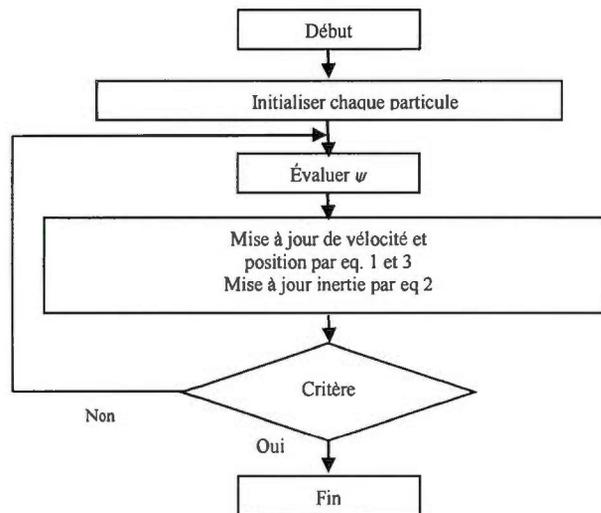
$$w(t) = w_{max} - (w_{max} - w_{min}) \cdot t / T_{max} \quad (2)$$

où  $t$  et  $T_{max}$  sont respectivement, l'instant (l'itération) en cours et le nombre maximum autorisé d'itérations. La position de chaque particule à la prochaine étape est ensuite évaluée comme la somme de sa position actuelle et de la vitesse obtenue par éq. (1) (le tout décrit dans la dimension  $d$ ):

$$p_{i,d}(t+1) = p_{i,d}(t) + v_{i,d}(t+1) \quad (3)$$

$$\text{pour } i=1, \dots, M \quad \text{et} \quad d=1, \dots, N$$

Ces opérations sont répétées pour  $T_{max}$  itérations ou jusqu'à ce qu'un autre critère d'arrêt soit vérifié. Un critère de convergence typique est constitué par la minimisation de la fonction objective  $\psi$  (la sémantique de la fonction étant dans ce cas celle de l'erreur que l'on cherche à minimiser). L'organigramme de la PSO standard est schématisé par la figure 2.4.



**Figure 2.4 Organigramme de la PSO standard.**

Ceci est l'approche générale dans la conception d'algorithme PSO. Cependant dans des implémentations plus sophistiquées (Norman, 2006), les particules ne sont pas homogènes et ont des buts spécifiques. Dans ce type d'implémentations, un groupe d'oiseaux préfère chasser alors qu'un autre groupe va préférer se percher. Ce type d'implémentations est plus robuste et puissant qu'une approche homogène simple, mais il rend plus complexe la conception de l'algorithme. Dans le cadre du travail proposé, la problématique à résoudre se situe dans la phase de rappel du cycle CBR et un essaim homogène suffit à la tâche.

La quantité de particules allouées à la résolution du problème dépend essentiellement de deux paramètres: la taille de l'espace de recherche et le rapport entre les capacités de calcul de la machine et le temps maximum de recherche. Il n'y a pas de règle pour déterminer ce paramètre, faire de nombreux essais permet de se doter de l'expérience nécessaire à l'appréhension de ce paramètre.

La topologie du voisinage définit avec qui chacune des particules va pouvoir communiquer. Il existe de nombreuses combinaisons dont (Clerc, 2005) :

- a) topologie en étoile : chaque particule est reliée à toutes les autres (l'optimum du voisinage est l'optimum global);
- b) topologie en anneau : chaque particule est reliée à  $n$  particules (en général,  $n = 3$ ), c'est la topologie la plus utilisée ;
- c) topologie en rayon : les particules ne communiquent qu'avec une seule particule centrale.

Le facteur d'inertie permet de définir la capacité d'exploration de chaque particule en vue d'améliorer la convergence de la méthode. Une grande valeur est synonyme d'une grande amplitude de mouvement et donc, d'exploration globale. Une faible valeur est synonyme de faible amplitude de mouvement et donc, d'exploration locale. Fixer ce facteur, revient donc à trouver un compromis entre l'exploration locale et l'exploration globale. La taille du facteur d'inertie influence directement la taille de l'hypermespace exploré et aucune valeur ne peut garantir la convergence vers la solution optimale.

Comme c'est le cas pour d'autres algorithmes en essaims, le choix des paramètres adéquat est important pour une exécution efficace de la PSO. Un travail considérable a été fait pour sélectionner une combinaison de valeurs qui fonctionnent bien dans un large éventail de problèmes. Ainsi, Clerc (2005) a donné quelques directives générales pour choisir la bonne combinaison. Il a proposé d'utiliser les valeurs suivantes:

- Taille  $M$  de l'essaim dans  $[20, 40]$ , avec une préférence pour 20 particules.
- Le paramètre cognitif  $c_1$  dans  $]0, 1[$ , avec une préférence pour la valeur 0.7.
- Le paramètre social  $c_2 \sim 1.5$  avec une préférence pour la valeur 1.43.

- La vitesse maximale  $V_{max} \sim (p_{max} - p_{min})/2$ , où  $p_{max}$  et  $p_{min}$  sont respectivement les valeurs maximale et minimale prises par la dimension dans l'espace de recherche.  $V_{max}$  seulement si  $c_1 > 1$  et peut être différent pour chaque dimension.

Engelbrecht (2007) explique que le choix du facteur d'inertie dans le cas de variations linéaires<sup>8</sup> comme représenté dans l'équation (2), peut prendre  $w_{max}=0.9$  et  $w_{min}=0.4$  comme valeurs initiale et finale.

Pour  $T_{max}$ , le nombre maximum d'itérations, il est généralement fixé à 1000 itérations, bien que des valeurs de l'ordre du dixième aient été déjà rencontrées lors de nos investigations.

Notons que les valeurs différentes de paramètres peuvent conduire à des résultats meilleurs ou moins bons selon le problème en main. Le meilleur moyen de régler ces paramètres est de procéder à une analyse de sensibilité dans le cadre de la description du problème.

## 2.6 Variantes et hybridations de la PSO

L'hybridation, dans les systèmes intelligents, vise à combiner les propriétés souhaitables des différentes approches visant à réduire leurs faiblesses individuelles. De multiples systèmes PSO hybrides ont été utilisés dans des contextes applicatifs spécifiques. Dans ce qui suit, nous passons brièvement en revue certaines de ces approches.

L'hybridation avec algorithmes évolutionnaires (AE), y compris les algorithmes génétiques (AG), a été une stratégie populaire pour l'amélioration des performances PSO. Les deux approches étant basées population, de telles hybridations sont facilement formulées. Angeline (1998) a appliqué un mécanisme de sélection du tournoi, afin de remplacer la vitesse et la position de chaque particule peu

---

<sup>8</sup> Le choix du facteur d'inertie dans le cas statique doit être fait en conjonction avec la sélection des valeurs pour  $c_1$  et  $c_2$ . Engelbrecht (2007) rapporte une valeur de:  $w > (c_1 + c_2)/2 - 1$ .

performante avec celles des particules les plus performantes. Ce mouvement dans l'espace améliore la performance dans trois des quatre fonctions de test utilisées, néanmoins il s'est éloigné de la métaphore sociale de la PSO.

Le développement du génotype dans la production du comportement du phénotype qui est influencé par l'environnement a été l'inspiration du modèle de cycle de vie proposé par Krink et Løvebjerg's dans (Krink et Løvebjerg, 2002). Les auteurs ont permis à chaque solution candidate de décider de la voie la plus productive à prendre (PSO, GA ou hill climber) en fonction de la connaissance de ses récents succès dans l'amélioration de sa fonction d'adaptation. Une fois la décision prise, l'individu se joint à une population de candidats qui partagent le même programme de développement. Dans des expériences utilisant des fonctions de test communes, le modèle du cycle de vie a dépassé les populations utilisant les techniques individuelles séparément.

Zhang et Xie dans (Zhang et Xie, 2003), ont également utilisé différentes techniques en tandem, dans leur PSO à évolution différentielle PSO (DEPSO). Dans ce cas, les opérateurs de PSO canonique et de DE (Differential Evolution) ont été utilisées en générations alternées. L'hybridation a été jugée réussie pour certaines fonctions tests, mais pas toutes. Les résultats indiquaient que DEPSO améliorerait la PSO dans la résolution des problèmes à plus grande dimension.

La mutation gaussienne a été combinée avec les règles de mise à jour de la vitesse et de la position par Higashi et Iba dans (Higashi et Iba, 2003). Elle a été testée sur les fonctions unimodales et multimodales. L'hybridation a obtenu de meilleurs résultats que ceux des algorithmes génétiques et de la PSO seuls.

La PSO coopérative de Van den Bergh et Engelbrecht dans (Van den Bergh et Engelbrecht, 2004) a pour objectif de minimiser l'augmentation exponentielle de la difficulté d'optimisation pour les problèmes à dimensions élevées en considérant chaque dimension comme un problème unidimensionnel. Plusieurs stratégies de

coopération ont été développées, où les petits essaims prennent en charge chaque dimension et une dimension transversale de communication permet à la solution globale de progresser vers l'objectif. Cependant, l'approche présente le risque potentiel de stagner en raison de l'évaluation de l'essaim en série et des sous-essaims de trouver des pseudominimums.

Liu et Abraham dans (Liu et Abraham, 2005) ont hybridé une PSO turbulente (TPSO) avec un contrôleur à logique floue pour produire une Fuzzy Adaptive TPSO (FATPSO). La TPSO a utilisé le principe selon lequel la convergence prématurée de la PSO est causée par des particules de stagnation sur une solution sous-optimale. Une vitesse minimale a été introduite avec le remplacement de la mémoire de vitesse par un opérateur de turbulence aléatoire lorsqu'une particule a dépassé. L'extension de la logique floue a été ensuite appliquée pour réguler de manière adaptative les paramètres de vitesse au cours d'une optimisation, permettant ainsi, une recherche exploratoire à grosse granularité de se produire dans les premières phases avant d'être remplacé par l'exploitation à granularité fine plus tard. La technique a été évaluée sur des problèmes avec haute et faible dimensionnalité et les résultats trouvés étaient bons dans les deux cas. Notamment, lorsque la performance de PSO canonique se dégrade considérablement face aux problèmes à dimensionnalité forte, la TPSO et FATPSO demeurent très peu touchées.

La PSO mémétique (MPSO), proposée par Petalas, Parsopoulos et Vrahatis (2007) est un algorithme hybride qui combine la PSO avec des techniques de recherche locale. La MPSO se compose de deux éléments principaux, un composant global qui est responsable de la recherche globale dans l'espace de recherche, et un composant local, qui raffine la recherche autour des solutions potentielles selon trois schémas possibles : La recherche locale est effectuée sur toutes les meilleures positions globales ou bien de manière aléatoire sur certaines meilleures positions ou bien une combinaison des deux premiers schémas. Les trois schémas peuvent être appliqués systématiquement à chaque itération ou bien durant certaines itérations uniquement.

Le travail de Mikki et Kishk (2008) vise à fournir une nouvelle vision de la PSO au moyen d'une analogie formelle avec les systèmes physiques. En postulant que le mouvement de l'essaim se comporte comme des particules classiques et quantiques, ils établissent un lien direct entre deux champs d'étude indépendants, l'optimisation et la physique. Dans ce cadre, il devient tout à fait naturel de dériver l'algorithme PSO quantique récemment introduit de la théorie hamiltonienne ou du lagrangien des systèmes. La théorie physique de la PSO est utilisée afin de suggérer des améliorations dans l'algorithme lui-même, comme les techniques d'accélération de la température et la théorie des conditions limites périodiques.

Dans un travail plus récent (Yin, 2010), Ying Tan présente un algorithme PSO utilisant un mécanisme de clonage (CPSO) inspiré du système immunitaire naturel. L'algorithme se caractérise par une convergence rapide et une mise en œuvre facile. En clonant le meilleur individu dans les générations successives, l'algorithme élargit la zone proche de la solution potentielle et accélère l'évolution de l'essaim, menant à une plus grande capacité d'optimisation et une convergence plus rapide que la PSO classique. Comme variante, on trouve la stratégie de l'avance-et-retraite appliquée pour trouver les minima proches dans un espace élargi de solutions, ce qui accélère considérablement la PSO avant la prochaine opération de clonage.

## 2.7 Situation de la PSO

Dans la mesure où le concept de 'swarm' avait été inspiré des automates cellulaires, il serait naturel de voir en quoi il s'en distingue. En effet, au même titre que les particules, les cellules obéissent à des règles simples menant à un phénomène émergent. Au moins trois des propriétés fondamentales que possèdent les automates cellulaires, à savoir : le parallélisme, la proximité (le nouvel état d'une cellule ne dépend que de son état actuel et de l'état du voisinage immédiat) et l'homogénéité (les lois sont universelles, c'est-à-dire communes à l'ensemble de l'espace de l'automate), existent aussi pour une population de particules. Il existe, pourtant, une différence

notable au niveau du fonctionnement, les automates cellulaires fonctionnent de manière *discrète* alors que la PSO a été réfléchi *continue* (même si des versions discrètes sont venues par la suite).

Une autre analogie s'impose lorsqu'on parle d'approches évolutionnistes. En effet, là aussi il est question de population d'individus évoluant selon des règles simples menant un phénomène émergent. Cependant, la PSO offre une souplesse de formulation alors que le codage dans les algorithmes évolutionnistes les rend rigides.

## 2.8 Conclusion

La PSO se caractérise par une cognition d'ordre éthologique donc une forme de cognition sociale très intéressante. En effet, la PSO s'inspire des essaims et mime donc un comportement d'animaux sociaux en vue de résoudre un problème. La PSO a été appliquée avec succès dans des domaines aussi diversifiés que l'apprentissage de réseaux de neurones, la classification, et l'affectation de tâche (Holden et Freitas, 2005; Norman, 2006).

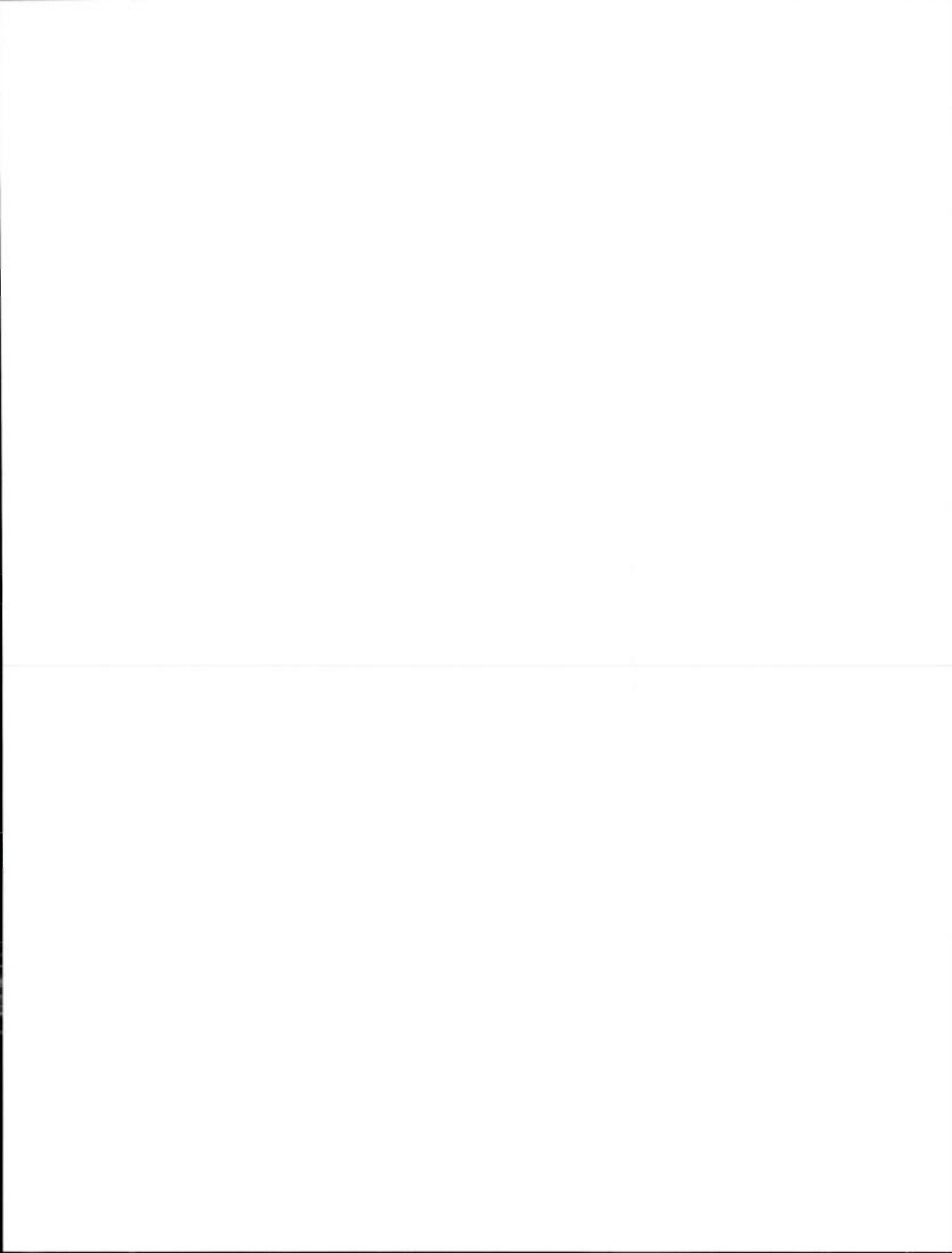
Tout au long de ce chapitre, nous avons vu les aspects théoriques de l'intelligence par essaims, et plus particulièrement la PSO. L'application usuelle de la PSO reste l'optimisation. À partir du moment où la sémantique du problème à résoudre peut s'exprimer sous forme d'une fonction à optimiser, la PSO s'applique. Dans la deuxième partie de ce travail, nous décrivons une application non usuelle des capacités d'optimisation de la PSO, il s'agit d'optimiser une fonction capturant la sémantique de similarité, puis de réutilisabilité par extension, dans le cadre d'une tâche de recherche en mémoire de cas.



## CHAPITRE III

# Un modèle de mémoire de cas basé PSO

1. Introduction
2. La mémoire de cas dans la littérature
3. À la recherche du bon cas
4. Architecture proposée
5. Modélisation
6. Conclusion



### 3.1 Introduction

Le processus de rappel en mémoire est fortement influencé par l'organisation de la mémoire et par les stratégies de récupération. Par conséquent, la précision et la vitesse de la tâche de rappel sont deux paramètres importants pour évaluer la performance le module de remémoration dans un système CBR.

Le CBR peut être combiné de manière synergique avec d'autres outils d'intelligence artificielle. Parmi les combinaisons possibles, nous faisons appel dans cette thèse à l'intelligence en essaim pour effectuer un rappel rapide dans une mémoire associative.

Dans ce chapitre, nous commençons par rappeler la situation de la mémoire de cas dans la littérature avant de proposer une architecture répondant aux exigences du module de remémoration.

### 3.2 La mémoire de cas dans la littérature

Il existe plusieurs organisations de la mémoire avec un algorithme de recherche associé. On peut les diviser comme suit :

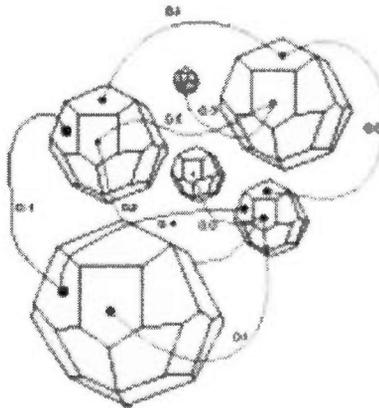
La mémoire plate/algorithme de force brute : les cas sont stockés de manière séquentielle dans une liste, un tableau simple ou un fichier (Kolodner, 1993). Les cas seront récupérés par application séquentielle d'une fonction d'appariement pour chaque cas dans la base en gardant trace d'un score d'appariement ; le cas avec le meilleur score est retourné. Il n'existe aucune organisation particulière des cas et l'algorithme de recherche est très simple puisque la fonction d'appariement fait tout le travail. L'avantage majeur de cette organisation de la mémoire est que la recherche est exhaustive. Par conséquent, la précision de la récupération est seulement une fonction de la fonction d'appariement. L'ajout d'un nouveau cas est peu coûteux, mais l'organisation de la mémoire devient coûteuse lorsque la base de cas est trop grande.

Pour remédier à cet inconvénient, il est possible d'utiliser des alternatives telles que les indices de surface pour réduire le nombre total de candidats, ou le partitionnement, ou encore des mises en œuvre parallèles (Kolodner, 1993).

La mémoire structurée/ algorithme basé index: les mémoires de cas sont riches de structures généralisées telles que les concepts, les prototypes et les cas abstraits. L'accumulation des généralisations ou des abstractions facilite l'évaluation des situations et permet un meilleur contrôle de l'indexation. Ces structures peuvent être organisées en hiérarchies conceptuelles, arbres de décision, taxonomies orientées objet, treillis de concepts formels, et B-arbres (Bichindaritz, 2006). Nous trouvons aussi dans cette catégorie les réseaux à caractéristiques partagées où les cas semblables sont regroupés ensemble (Kolodner, 1993). Des hiérarchies sont formées lorsque ces clusters sont subdivisés en petits groupes. De manière duale, on trouve les réseaux de discrimination où une discrimination de cas est faite comme conséquence du clustering dans les réseaux à caractéristiques partagées (Bartsch-Spörl et al., 1999). Une variante de cette approche offrant une solution au problème d'information manquante, mais qui reste coûteuse à mettre en œuvre en les réseaux de discrimination redondants (Wiltgen, Goel et Vattam, 2011)

Tous les modèles de la mémoire qui utilisent une recherche descendante (*Top/Down*) partagent deux caractéristiques désirables (Lenz et al., 1998): structuration des données en regroupant les objets liés et une recherche efficace par des algorithmes de recherche d'arbre traditionnels. Cependant, il y a des limites à leur usage, dont le parcours de la mémoire en répondant à une séquence ordonnée de questions au niveau des nœuds internes (dans le cas d'informations incomplètes, ceci pourrait conduire à des chemins erronés), ainsi que l'accès difficile à des clusters voisins avec des cas semblables lorsqu'ils atteignent un cluster au niveau d'une feuille de l'arbre. Deux exceptions notables dans la catégorie des approches basées indice sont le modèle Fish & Shrink (Schaaf, 1996) et le modèle CRN (Lenz et al., 1998).

Shaaf (1996), introduit une structure de données basée aspects pour les représentations des cas et des liens pour stocker les similarités entre les aspects. Supposons deux cas considérés comme similaires dans un domaine donné et selon un aspect précis. Si l'on change l'aspect considéré, ces deux cas deviennent dissimilaires. Malheureusement, lors de la construction des bases de cas, il n'est pas aisé de changer de point de vue, car l'on utilise des similarités statiques pour organiser la base de cas. L'idée de l'algorithme de Fish & Shrink est de considérer le cas comme étant une structure supportant de multiples facettes ou aspects. La similarité de l'ensemble des cas de la base sera calculée selon chacun des aspects du cas et stockée préalablement. Lors de la recherche, on peut sélectionner les aspects à considérer et éventuellement les combiner selon leurs poids si l'on doit prendre en compte plus d'un aspect. Ainsi, la base de cas peut être vue comme un réseau de cas (voir figure 3.1). Où un polyèdre représente un cas.

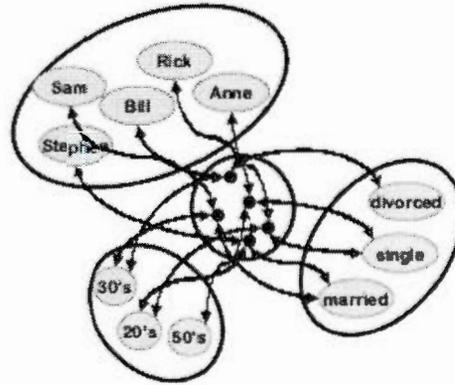


**Figure 3.1** La base de cas vue comme un réseau de cas (extraite de (Schaaf, 1996)).

Une facette du polyèdre représente un aspect et l'étiquette d'un arc reliant deux cas, représente la distance calculée entre ces cas selon l'aspect considéré. Deux cas sont dits voisins selon un aspect s'ils sont reliés par un arc selon cet aspect. Les arcs

peuvent être orientés si la distance n'est pas symétrique. L'auteur explique comment les changements de point de vue sur les cas peuvent être considérés comme une combinaison spontanée et pondérée des aspects. Cela permet une redéfinition de la similarité des cas en fonction du contexte en utilisant uniquement des calculs à coûts peu élevés. Ce modèle présente des avantages tels que : le calcul flexible de distance pour la requête, différentes tâches de recherche peuvent être effectuées, l'efficace, puisque différents calculs de distance peuvent être sauvegardés de plus il peut être utilisé comme solution « anytime ». Il est parfaitement convenable lorsque l'application nécessite des calculs de similarité coûteux (ex : représentation de graphes). Néanmoins, il a les inconvénients suivants : toutes les distances entre aspects de cas doivent être calculées au préalable et la fonction distance doit satisfaire l'inégalité triangulaire.

Lenz et al. proposent les CRNs (pour Case Retrieval Nets) comme modèle de mémoire (Lenz et al., 1998). Ce modèle utilise les mécanismes de propagation d'activation et de complétion de l'information. Le modèle CRN offre une accélération de la remémoration significative par rapport à la recherche linéaire et a été déployé avec succès sur des bases de grande envergure avec 200 000 cas. Les fondements des CRNs sont inspirés des techniques de réseaux neuronaux et des modèles de mémoires associatives. Ce principe (Figure 3.2) s'écarte de la représentation classique des connaissances en mettant en commun, toute donnée susceptible d'être partagée par plusieurs objets. Exemple : Bill et Sam ont chacun 30 ans. La valeur 30 ans n'est représentée qu'une fois dans le réseau tout en étant utilisable aussi bien pour Sam que pour Bill.



**Figure 3.2** Principe d'une mémoire associative (extraite de (Lenz, 1999))

Les objets fondamentaux dans le contexte CRN sont les entités d'information (IE) qui représentent des éléments de connaissance de plus bas niveau de granularité (Lenz, 1999). Un cas est un ensemble d'entités indépendantes et une mémoire de cas est un réseau de nœuds correspondant aux IEs observées dans le domaine et des nœuds cas pointant sur un cas particulier. Les nœuds IEs sont reliés par des arcs de similarité pondérés par la valeur d'une fonction de similarité. Un nœud cas, est joignable à partir de ses nœuds IEs à travers des arcs de pertinence activés par une fonction binaire.

Parmi leurs caractéristiques remarquables (Lenz, 1999), un processus de rappel qui n'utilise pas de parcours d'arbre. En CRN, la remémoration est faite d'une manière reconstructive en récupérant les entités d'information qui serviront progressivement à la reconstruction du cas.

Le tableau 3.1 récapitule les méthodes d'extraction ci-dessus et indique leur complexité de calcul. Notez que la complexité de calcul est relative à l'étape de recherche et ne comprend pas le coût de construction de la structure de la mémoire. Ce coût pourrait être très important, surtout pour les deux dernières techniques.

**Tableau 3.1 Aperçu des méthodes de recherche en mémoire de cas (adapté de (Richter, 2004) et complété)**

Type	Méthode	Utilisation	Complexité computationnelle
À force brute	Recherche séquentielle	Bases de petite taille, Similarité simple	Dépend de la taille $n$ de la base de cas : $O(n)$
	Kd-Tree	Peu attributs, Grandes bases de cas	Dépend de la profondeur de l'arbre $O(\log_2 n)$
Basée Index	Fish & Shrink	Similarité complexe, Petites bases de cas	Dépend de la connectivité des aspects et de la taille de la requête
	CRN	Peu d'attributs numériques, Grandes bases de cas	Dépend de la taille de la requête, de la connectivité des entités d'information et de leur spécificité.

Dans l'ensemble des familles de modèles vus, la structuration de la mémoire est une étape incontournable. Cette structuration est basée sur un critère unique (généralement, la similarité) qui induit une perception particulière et contrainte par le critère choisi. Alors que le rôle du critère devrait être le filtrage lors de l'étape de sélection, on voit que ces modèles l'impliquent dans la structuration même de la mémoire. Il intervient donc comme outil de représentation de connaissances lors de la structuration de la mémoire. Or une bonne représentation des connaissances se doit être la plus proche de la réalité et non sensible aux outils (ici le critère). À cela s'ajoute le coût de calcul parfois prohibitif pour mettre en place cette organisation (comme c'est le cas pour le modèle Fish & Shrink). Dans la suite de ce chapitre, nous proposons un modèle de mémoire qui a la particularité de ne pas nécessiter une organisation préalable de la mémoire de cas et qui peut mener à :

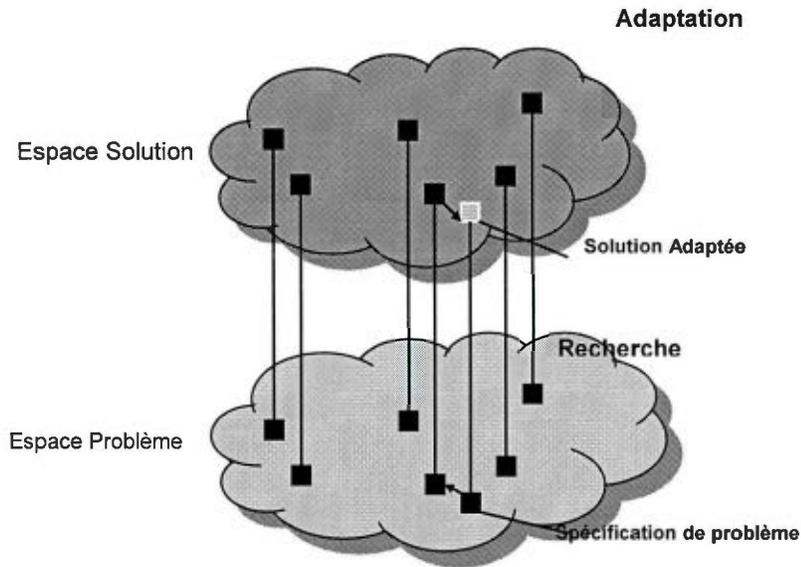
- Une réduction de l'effort de calcul lors de la construction de la mémoire de cas,
- Une amélioration de la rapidité et de la précision de remémoration puisqu'il n'y a ni phase de préfiltrage des cas de la base ni biais structurel nous forçant à emprunter tel ou tel chemin.

L'approche repose sur une vision non figée de l'organisation de la mémoire de cas. Dépendamment de la sémantique associée au critère de parcours, la mémoire de cas est investiguée différemment. Par conséquent, ce n'est plus la structure — figée au moment de la construction — qui guide la recherche, mais le critère de recherche — qui peut être variable au besoin.

### 3.3 À la recherche du bon cas

Toutes les étapes de l'inférence en CBR gravitent au tour de la base de cas et y retrouver la bonne expérience mémorisée joue un rôle crucial. Nos énoncés de thèse à ce propos étaient que : *1. La clé du succès de la résolution de problème de second principe appliqué au CBR est la remémoration du bon cas ; 2. Par bon cas, on entend celui qui rendra la réutilisation optimale, donc le plus similaire dans sa partie problème et le plus adaptable dans sa partie solution.*

Pour trouver le bon cas, il est intéressant de considérer que le CBR fonctionne sur deux espaces de symboles distincts, un espace de spécification ou de description de problème et un espace de description de solution (figure 3.3). La recherche de cas opère dans l'espace problème en tentant de trouver des appariements entre la description du problème cible et la partie problème des cas sources considérés. L'adaptation, quant à elle, opère dans l'espace solution en procédant à des transformations sur les parties de la solution du cas source retenu pour satisfaire le problème cible.

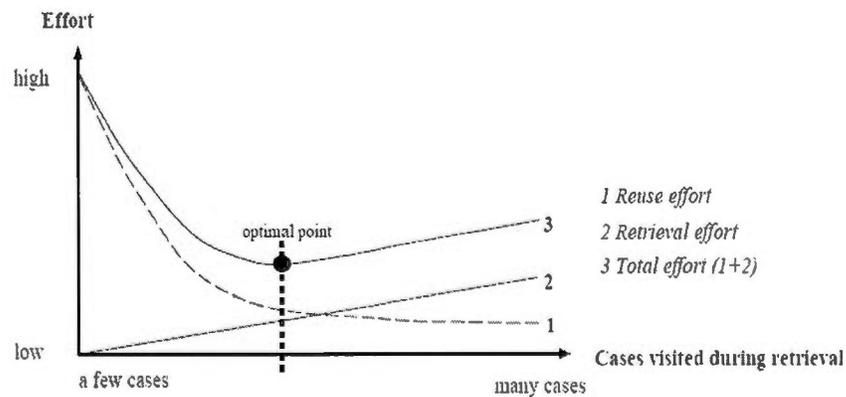


**Figure 3.3** Espace problème/Espace solution (adaptée de (Watson, 1997))

Dans cette perspective, la remémoration est l'exploration de l'espace problème en vue de trouver le bon ensemble d'appariements entre les spécifications du cas cible et celles des cas sources; l'adaptation est l'exploration de l'espace solution en vue de trouver le bon ensemble de transformation de la solution du cas source sélectionné en une solution du cas cible.

L'une des hypothèses majeures faites en Intelligence Artificielle est que les expériences similaires peuvent guider le raisonnement futur, la résolution de problème ou l'apprentissage; c'est ce que Smyth et Keane (1999) ont qualifié d'*Hypothèse de Similarité*. Cette hypothèse semble être la fondation des approches conventionnelles de recherche de cas, ces dernières faisant une séparation nette entre les étapes de recherche et d'adaptation avec l'idée est que la similarité dans l'espace problème est suffisante pour prédire la réutilisabilité future d'un cas. En ignorant ainsi le lien entre l'espace problème et l'espace solution, nous ignorons aussi toute

communication réelle entre l'étape de recherche et l'étape d'adaptation. Or l'objectif de la recherche est de fournir un cas candidat à la réutilisation qui optimise l'effort *total* (point optimal de la figure 3.4). De ce fait, le meilleur critère de choix du candidat devrait se résumer plutôt à : « *retrouver le cas le plus similaire dans sa partie problème et le plus adaptable dans sa partie solution* ». Ceci revient à l'intégration de deux critères de recherche en même temps, la similarité et l'adaptabilité, c'est-à-dire l'intégration des deux espaces problème et solution.

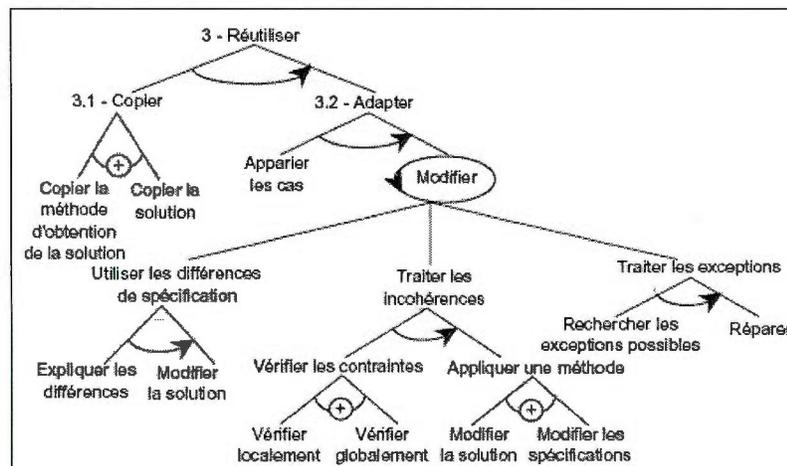


**Figure 3.4** Compromis entre recherche et réutilisation (extraite de (Bergmann et al., 1998))

La tâche de réutilisation (figure 3.5) commence par copier soit la solution, soit la méthode d'obtention de la solution du cas remémoré, et procède ensuite à l'adaptation. L'effort à consentir pour effectuer cette tâche pourrait être considéré comme indicateur de la réutilisabilité d'un cas au moment de la remémoration et guider ainsi ce processus en anticipant sur les besoins du processus de réutilisation. Cet effort est exprimé dans l'équation de transfert suivante :

$$\text{Coût\_Transfert}(s, c) = \text{Similarité}(s, c) * \text{Coût\_Copie}(s, c) + \text{Dissimilarité}(s, c) * \text{Coût\_Adaptation}(s, c) \quad (5)$$

Il s'agit du coût de transfert des connaissances du cas source  $s$  remémoré vers le cas cible  $c$ . La similarité entre cas source et cas cible est calculable soit par des métriques ou des heuristiques. La dissimilarité est la notion duale. Lorsque les problèmes source et cible sont identiques (similarité maximale = dissimilarité nulle), la réutilisation se limite à la tâche de copie, et il n'y aura pas d'adaptation; seul le terme en similarité agira, le terme en dissimilarité étant annulé.



**Figure 3.5** Décomposition de la tâche 'Réutiliser' (extraite de (Fuch, 1997))

D'un autre côté, lorsque le problème source présente des dissimilarités avec le problème cible sur tous les descripteurs, la dissimilarité est maximale et le coût d'adaptation sera calculé sur l'ensemble des descripteurs; l'effort d'adaptation sera donc plus important

L'appréciation de l'adaptabilité, qui est la potentialité d'un cas à être adapté, va être estimée à travers un 'coût d'adaptabilité' noté  $\mu()$ . Il est défini comme étant l'approximation de l'effort à fournir pour adapter une solution. Son calcul est basé sur les relations de dépendances présentes entre partie problème et partie solution du cas

source. C'est un score établi à partir des liens problème /solution d'un cas en considérant uniquement les parties problèmes concernées par la dissimilarité. On a :

$$\mu(c) = \sum_i \mu(p_i) \quad (6)$$

où  $p_i$  est une partie concernée par l'adaptation et  $\mu(p_i)$  est le nombre d'opérateurs nécessaires à son adaptation. L'appréciation de ce score est faite dans une étape de prétraitement de la base de cas.

Pour illustrer cette notion, nous présentons un exemple extrait de la base GUIDIET que nous verrons dans la partie 3 de ce document. Il s'agit d'une base de cas utilisée dans une application permettant de proposer un menu diététique à une personne selon son profil, ce dernier étant spécifié par les d'indicateurs suivants : âge, sexe, dépense énergétique, IMC (indice de masse corporelle) et maladies. La recommandation est faite en adaptant des cas constitués d'anciens profils accompagnés de leurs menus. L'adaptation se fait sur des règles établissant le lien entre un descripteur problème tel qu'IMC et les composantes du menu, par exemple :

Si (IMC>29) Alors Substituer [Féculent. Pain au chocolat] par (Féculent. Pain blanc)].

L'analyse de l'ensemble de ces heuristiques permet d'établir tous les scores de dépendance entre descripteurs problème et descripteurs solution. Ainsi, l'expression

$$\mu(\text{IMC} > 29) = 6$$

signifiera, par exemple, qu'un changement du descripteur problème IMC>29 va induire l'utilisation d'opérateurs d'adaptation sur 6 descripteurs solution (féculents, produits laitiers, matière grasse, viande/Œuf, poissons et divers : Pizza, Sandwich, ...)

### 3.4 Architecture de mémoire proposée

L'efficacité du processus de rappel en mémoire de cas se traduit en termes de stratégie d'exploration d'un espace de recherche. Bien entendu, l'exhaustivité d'un parcours linéaire place ce dernier au premier rang de la précision, mais au risque d'un temps de parcours excessif. Quant aux approches s'appuyant sur une structuration préalable de l'espace, elles gagnent en temps, mais perdent en précision ; sans oublier l'impact potentiel du biais structurel introduit dans la perception de l'espace et, par là même, la généralité des algorithmes pour différents domaines applicatifs. La question revient donc à dire comment parcourir l'espace de recherche sans le structurer au préalable? Notre énoncé de thèse (énoncé 3) à ce niveau était : *L'élimination du biais structurel améliore la perception de l'espace de recherche et la qualité du rappel par là même.*

En effet, si l'on se réfère à la figure 3.3, la séparation des espaces de description du problème et de la solution se traduit, dans les approches CBR classiques, en un espace problème structuré afin de faciliter l'accès à l'espace solution. Cependant, le travail sur un espace unique<sup>9</sup>, intégrant problème et solution, rend les approches conventionnelles de recherche en mémoire de cas inopérantes, puisqu'elles se basent sur des indices de surface issus de l'espace problème uniquement. Ce qui est demandé est une approche qui permet l'exploration rapide et efficace de l'espace de recherche sans structuration préalable et pouvant garantir l'intégration des critères de recherche similarité et adaptabilité. Une telle approche peut être réalisée par l'optimisation par essaims de particules (PSO).

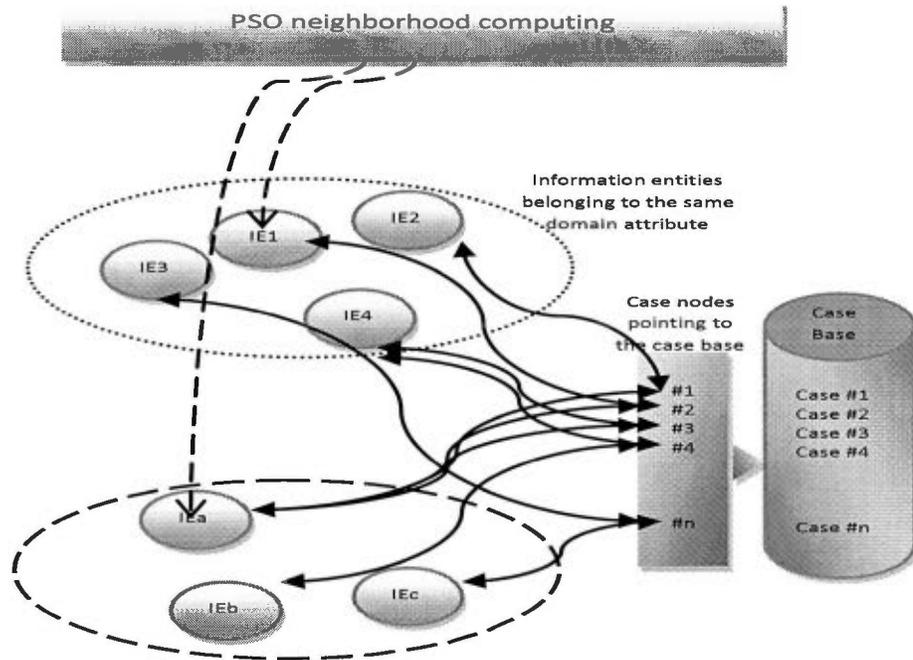
---

<sup>9</sup> Notons que le fait de travailler sur un espace unique ne change en rien la représentation des connaissances au niveau du cas puisque ce sont toujours les mêmes descripteurs (appartenant à l'espace problème et à l'espace solution) qui sont utilisés. Ce qui est impacté par contre, ce sera la structure de la mémoire lorsqu'elle a besoin de cette distinction (ce qui est le cas pour les approches classiques).

L'option de l'exploration stochastique semble avoir apporté ses fruits dans divers domaines de l'intelligence artificielle (Engelbrecht, 2007). Nous proposons son exploitation dans le cadre de la recherche en mémoire. L'idée est donc de réaliser un parcours stochastique de l'espace de recherche sans s'appuyer sur une structure de parcours préalable et de manière que la cible soit identifiée par ses caractéristiques de similarité et d'adaptabilité. L'énoncé de thèse (énoncé 4) formulé à ce niveau était : *le parcours stochastique de l'espace de recherche permet la génération du voisinage caractérisé par une réutilisabilité forte puis un accès direct à ce voisinage dans une mémoire associative de cas.*

La nouvelle vision du problème de rappel en mémoire qui est proposée s'appuie sur la construction d'un voisinage du problème à résoudre. Le processus de rappel de cas  $y$  est formulé comme étant la génération d'un sous-espace de cas selon un critère spécifique : similarité au problème à résoudre et adaptabilité de la solution. Nous appelons ce sous-espace voisinage du problème cible et il est calculé par une stratégie de recherche PSO. Au final, il s'agit d'un ensemble de descriptions de cas générées artificiellement par PSO et qui seront candidates à l'appariement.

Nous partons d'une structure plate de la mémoire de cas et construisons une structure associative avec deux types de nœuds: les nœuds 'cas' et les nœuds 'valeur'. Un nœud 'valeur', ou nœud 'Entité d'Information' (EI), représente une valeur particulière pour un attribut problème (voir figure 3.6). Il est lié à tous les nœuds cas où il apparaît. Le nœud de cas souligne l'emplacement dans la base de cas où le cas en entier est stocké; c'est le principe de mémoire accessible par le contenu. Dans la figure 3.6 par exemple, l' $IE_2$  et l' $IE_a$  pointent vers le nœud-cas#1 signifie que le cas #1 de la base, qui est pointé par le nœud-cas#1, a  $IE_1$  comme valeur du premier attribut et  $IE_a$  comme valeur du deuxième attribut.



**Figure 3.6** Architecture de mémoire proposée.

La particularité de la structure proposée est que le cas est accessible par son contenu. Une autre particularité est que la structure pourrait facilement et automatiquement être construite par simple compilation de la mémoire de cas et la construction de listes de nœuds de valeur et les nœuds de cas au cours d'une phase de prétraitement. Par ailleurs, cette construction ne présente aucun biais, puisqu'elle n'est reliée à aucun critère organisationnel. Elle ne fait que mettre en relations le contenu  $\{IE_i\}$  et les contenants  $\{\text{nœuds-cas}\#i\}$  et reflète la réalité dans son intégralité.

Ainsi chaque domaine de descripteur est représenté par un ensemble de nœuds valeurs. Chaque nœud-cas est pointé par un ensemble de nœuds valeurs, chacun d'eux appartenant à un domaine de descripteur spécifique. Chaque nœud-cas pointe sur le cas correspondant dans la base de cas. Par exemple une requête spécifiant le sous-ensemble de nœuds valeurs =  $\{IE_4, IE_a\}$  aura comme résultat le cas pointé par les nœuds-cas appartenant à  $\{\#3, \#4\} \cap \{\#1, \#2, \#3\} = \{\#3\}$

Nous utilisons l'algorithme PSO pour déterminer le voisinage d'un nouveau problème. L'espace de recherche est parcouru avec le but de minimiser une fonction objective exprimant une sémantique de réutilisabilité dans le contexte du problème cible. Ainsi, à la fin du processus, la solution calculée constitue un voisinage artificiel du problème cible qui comprend tous les cas potentiellement réutilisables dans le contexte du problème posé. Il sera fourni en entrée du dispositif d'accès à la mémoire. Chaque problème source calculé par le module PSO sera pointé directement dans la base de cas via le réseau (voir Figure 3.6). L'étape d'extraction est basée sur la description du problème seulement, mais pourrait également inclure des descripteurs de la solution pour contraindre encore plus le processus de remémoration.

À présent, nous mettons le focus sur le module PSO pour le calcul du voisinage, et proposons dans ce qui suit, une démarche pour construire le voisinage du problème cible.

### 3.5 Modélisation

#### 3.5.1 Mise en équation

Rappelons<sup>10</sup> que la PSO est fondée sur un essaim de  $M$  individus appelés particules, chacune représentant une solution à un problème à  $N$  dimensions. Son génotype est composé de  $2 \times N$ , les  $N$  premiers représentant les coordonnées des particules dans l'espace de recherche, et les  $N$  derniers, les composantes de leur vitesse.

La vitesse  $v_{i,d}(t+1)$  de la  $i^{\text{ème}}$  particule pour la dimension  $d$ , à la prochaine étape  $t+1$  est:

$$v_{i,d}(t+1) = w \cdot v_{i,d}(t) + c_1 \cdot U(0, 1) \otimes (b_{i,d}(t) - p_{i,d}(t)) + c_2 \cdot U(0, 1) \otimes (b_{g,d}(t) - p_{i,d}(t)) \quad (1)$$

<sup>10</sup> Toutes ces notions ayant été définies au chapitre 2, nous faisons un bref rappel des équations utilisées afin d'introduire de nouvelles notions.

$$i=1, \dots, M \text{ et } d=1, \dots, N$$

Une amélioration de la PSO originale est de faire varier  $w$  au cours de l'exécution, à partir d'une valeur maximale  $w_{max}$ , de le décrémenter linéairement à mesure que le nombre d'itérations augmente jusqu'à une valeur minimale  $w_{min}$  comme suit:

$$w(t) = w_{max} - (w_{max} - w_{min}) \cdot t / T_{max} \quad (2)$$

La position de chaque particule à la prochaine étape est ensuite évaluée comme la somme de sa position actuelle et de la vitesse obtenue par éq. (1):

$$p_{i,d}(t+1) = p_{i,d}(t) + v_{i,d}(t+1) \quad (3)$$

$$i=1, \dots, M \text{ et } d=1, \dots, N$$

Le calcul est effectué afin d'optimiser une fonction objective  $\psi$ . La sémantique attribuée à cette fonction dépend du problème à résoudre comme nous le verrons à travers différents exemples au chapitre 5 où la distance euclidienne est notamment utilisée et l'exemple de l'équation 5 au présent chapitre. Ces opérations sont répétées pour  $T_{max}$  itérations ou jusqu'à ce qu'un autre critère d'arrêt exprimé en fonction de  $\psi$  soit vérifié. Ceci est l'approche générale dans la conception d'algorithme PSO. L'algorithme PSO peut être résumé par le pseudo-code suivant:

---

#### *Algorithme PSO Standard (gbest)*

---

- (1) *début*
- (2) *pour chaque particule i*
- (3) *initialiser position et vitesse*
- (4) *fin-pour*
- (5) *Tant que (non nombre maximum d'itérations atteint)*
- (6) *pour chaque particule i*
- (7) *déterminer valeur de la fonction objective  $\psi_i(t)$*
- (8) *Si  $\psi_i(t)$  meilleure que  $\psi(b_i(t))$*
- (9) *alors  $\psi(b_i(t)) = \psi_i$  et  $b_i(t) =$  position courante*
- (10) *Si  $\psi_i(t)$  meilleure que  $\psi(b_g(t))$*
- (11) *Alors  $\psi(b_g(t)) = \psi_i(t)$  et  $b_g(t) =$  position courante*

- (12) *fin-pour*  
 (13) *pour chaque particule i*  
 (14) *calculer la vitesse selon équation (1)*  
 (15) *mettre à jour la position de la particule selon l'équation (3)*  
 (16) *fin-pour*  
 (17) *mettre à jour le facteur d'inertie selon l'équation (2)*  
 (18) *fin-tant que*  
 (19) *fin*
- 

En plus de l'algorithme PSO de base, il existe des implémentations plus sophistiquées qui utilisent des mécanismes de mise à jour de la position différents. Parmi lesquels, deux sont d'intérêt pour ce travail: le confinement et la dispersion du vent.

L'action du mécanisme de confinement est de limiter les changements de position à un intervalle (Clerc, 2005). Il consiste à borner les composants de la position d'une particule de telle sorte que, pour la  $d^{\text{ème}}$  composante de position dans l'espace à  $N$  dimensions, nous avons:

$$p_{i,d}(t+1) = \text{MIN}(\text{MAX}(p_{i,d}(t) + v_{i,d}(t+1), P_{dmin}), P_{dmax}) \quad (3a)$$

Où  $P_{dmin}=0$  et  $P_{dmax}=1$  pour la dimension  $d$

Le second mécanisme, décrit dans (Chandramouli et Izquierdo, 2006) comme une approche chaotique, est la dispersion par le vent. La vitesse et la direction du vent sont considérées pour modéliser l'atmosphère biologique au moment de la mise à jour des positions de particules. La mise à jour de la vitesse du vent est donnée par l'équation suivante:

$$vw(t+1) = vw(t) + v_{op} \cdot \text{rand}() + v_{su} \cdot \text{rand}() \quad (4)$$

Où  $vw$  est la vitesse du vent,  $v_{op}$  est le facteur de la direction opposée égal à -1 et  $v_{su}$  le facteur de la direction support égal à 1;  $\text{rand}()$  étant un vecteur généré aléatoirement. La vitesse du vent peut avoir l'un des effets suivants: opposition ou support au mouvement de la particule. Le facteur d'opposition ralentit la particule dans son sa lancée vers la meilleure position globale, alors que l'effet du facteur de support est l'augmentation de sa vitesse pour atteindre la meilleure position globale.

Chaque particule est séparément mise à jour par l'équation du vent. Ce qui est dû au fait que les particules sont spatialement séparées les unes des autres, et sont ainsi soumises à différentes forces dynamiques de l'atmosphère. Lorsque les valeurs des facteurs 'opposé' et 'support' au vent sont égales, une atmosphère statique est modélisée. L'équation de mise à jour de la position pour la dimension  $d$  dans l'espace à  $N$  dimension est donnée par:

$$p_{i,d}(t+1) = p_{i,d}(t) + v_{i,d}(t+1) + vw_d(t+1) \quad (3b)$$

Lorsque combinée au mécanisme de confinement, nous avons:

$$p_{i,d}(t+1) = \text{MIN}(\text{MAX}(p_{i,d}(t) + v_{i,d}(t+1) + vw_d(t+1), P_{dmin}), P_{dmax}) \quad (3c)$$

Les valeurs initiales de la vitesse du vent pour une direction donnée jouent un rôle important dans la convergence finale des particules vers la solution optimale. En outre, ce paramètre assure la recherche optimale de l'espace solution.

### 3.5.2 La PSO à valeurs mixtes

#### 3.5.2.1 Problème

Travailler avec un algorithme PSO pur implique un espace de recherche continu. Or pour des problèmes du monde réel, l'espace de description du problème peut contenir des descripteurs à valeurs continues, discrètes et/ou symboliques.

Les données mixtes sont fréquentes dans les tâches de Résolution de problème. Par exemple, lors du tri des patients dans un hôpital, afin de décider s'il y a lieu de les admettre ou de les traiter en ambulatoire. Le résultat repose généralement sur les descripteurs qui peuvent être continus (par exemple la température), nominaux (par exemple la présence ou l'absence de certains symptômes) et des mesures ordinales (score par exemple). Un autre exemple est l'analyse d'une demande de crédit dans un contexte bancaire. L'évaluation de la demande se base sur des informations telles que l'âge (discret), le gain annuel (continu) ou le statut marital (catégoriel).

Lorsqu'on traite des données mixtes, une manière typique de procéder passe par à une étape préliminaire de codage qui apparie des valeurs non numériques à des nombres entiers, mais elle se heurte à deux problèmes connexes: 1) Comment établir une relation d'ordre sur les données transformées ? 2) Comment remédier au biais causé par une représentation des connaissances plate (non pondérée) qui a perdu l'importance relative des différentes valeurs non numériques pendant le processus de codage.

### 3.5.2.2 Solutions proposées dans la littérature PSO.

Plusieurs méthodes ont été adoptées pour discrétiser l'algorithme PSO de sorte qu'il puisse être utilisé pour résoudre des problèmes discrets pour lesquels il n'est pas possible de faire "voler" des particules en continu.

Le codage en chaîne binaire est une modification de l'algorithme PSO pour résoudre les problèmes avec des éléments de solution à valeurs binaires. Il a été développé par les créateurs de PSO dans (Kennedy et Eberhart 1997). La signification de la variable vitesse a été modifiée pour indiquer la probabilité de l'élément de solution correspondant avec une valeur de 0 ou 1. La vitesse est mise à jour de la même manière que pour la PSO classique. Aucun coefficient d'inertie n'est utilisé. La position de la particule est définie de façon aléatoire avec la probabilité de choisir 1, donnée par une fonction sigmoïde.

De multiples variantes de cette PSO discrète (ci-après dénommée DPSO) sont proposées. Al-Kazemi et Mohan (2002) utilisent une technique où les particules sont alternativement influencées par leur meilleure position et la meilleure position parmi leurs voisins. Dans cette stratégie DPSO, la vitesse est mise à jour comme dans la PSO continue, à la différence que les coefficients cognitif et social ne prennent que les valeurs 1 et -1, et seulement un nombre donné de combinaisons des coefficients différents sont possibles. Les combinaisons sont considérées comme des phases des particules et déterminent leurs directions de déplacement. À un moment donné,

chaque particule est dans l'une des phases possibles, et la phase suivante à sélectionner est déterminée au moyen de la phase précédente et le nombre d'itérations exécutées jusqu'à présent.

Dans (Yang, Wang et Jiao 2004), les auteurs ont développé un algorithme DPSO qui prend en compte un plus grand nombre de combinaisons des coefficients, appelés états quantiques des particules, et une équation de mise à jour de la vitesse légèrement différente, inspirée des principes du calcul quantique. Notons que cette méthode aussi est limitée à des problèmes discrets avec valeur binaire des variables.

Pampara et al. (2005) avaient mis au point une DPSO indirecte par réduction d'un problème binaire à une fonction trigonométrique continue ne comportant que quatre paramètres d'optimisation. Cette réduction est obtenue par des moyens de modulation angulaire, une technique couramment utilisée dans le traitement du signal et de télécommunications. L'algorithme standard PSO est alors appliqué pour optimiser les quatre paramètres de la fonction continue trigonométrique. La fonction est successivement échantillonnée à intervalles réguliers pour produire une valeur pour chaque intervalle continu. Si la valeur est positive, la valeur du bit affecté à l'intervalle correspondant prend 1, sinon la valeur du bit correspondant prend 0. L'ensemble de toutes les valeurs de bits générées associées aux intervalles représente le vecteur solution au problème binaire d'origine.

Pour traiter des données numériques discrètes, Omran et al. (2005) proposent d'arrondir les valeurs des particules continues à l'entier le plus proche afin de générer la solution discrète.

Hassan et al. (2009) proposent de convertir les positions à valeurs continues sous forme discrète au moyen d'une équation à l'origine utilisée pour confiner les ensembles de données dans un intervalle donné pour les traiter par réseaux de neurones. Ainsi, l'optimisation est réalisée avec des valeurs continues, mais des

valeurs discrètes sont utilisées pour résoudre la fonction d'adaptation sous sa forme discrète.

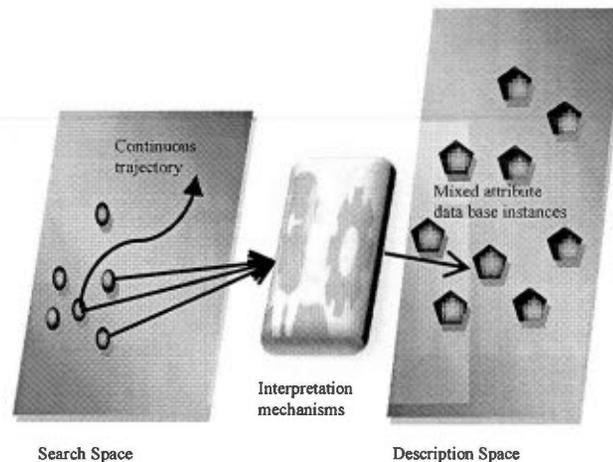
La PSO Multi-valuée (MVPSO) (Pugh et Martinoli, 2006) considère aussi les variables à valeurs discrètes. Tandis que la position de chaque particule est une matrice unidimensionnelle en PSO continu et un tableau à 2 dimensions en DPSO, dans MVPSO, elle est exprimée au moyen d'un tableau à 3 dimensions, où l'élément de la case  $i,j,k$  représente la probabilité que la  $i^{\text{ième}}$  particule dans l'itération  $j$  prend la valeur  $k$ . Pour évaluer la fonction d'adaptation d'une particule, les éléments de la solution sont générés de manière probabiliste avec une distribution sigmoïdale.

La Jumping PSO (JPSO) est une DPSO proposée dans (Moreno-Pérez et al. 2007), et étendue dans (Consoli et. 2009.). L'approche ne considère aucune vitesse puisqu'il y a absence de continuité de mouvement dans un espace discret, mais l'attraction par les meilleures positions est conservée. La JPSO considère un essaim de particules dont les positions évoluent dans l'espace solution, en sautant d'une solution à l'autre. Pour chaque itération, chaque particule a un comportement aléatoire ou guidé par l'effet d'un attracteur. L'algorithme prend en compte trois attracteurs pour le mouvement d'une particule: sa position propre meilleure à ce jour, la meilleure position de son voisinage social, interprétée comme la meilleure position obtenue au sein de l'essaim dans l'itération courante, et la meilleure position à ce jour obtenue par toutes les particules. Un saut qui s'approche d'un attracteur consiste à modifier une caractéristique de la solution actuelle par une caractéristique de l'attracteur. Chaque particule est en outre autorisée à se comporter de façon aléatoire en effectuant des sauts aléatoires. Il s'agit de sélectionner de manière aléatoire une caractéristique de la solution à en changer la valeur. Une inspiration naturelle pour ce processus se trouve dans les grenouilles qui sautent d'une feuille de nénuphar à l'autre.

Il est important de noter qu'aucune des solutions proposées n'est conçue pour traiter les données mixtes ensemble.

### 3.5.2.3 Solution proposée

Notre objectif est une approche permettant de traiter les données mixtes sans les séparer. Nous proposons de travailler avec deux espaces (figure 3.7): un espace de recherche où les particules évoluent de manière continue selon les lois de base de la PSO (Kennedy et Eberhart, 1995), et un espace de description qui reflète la réalité et où les entités peuvent avoir des valeurs mixtes. Il s'ensuit que si une particule évolue le long des axes continus dans l'espace des particules, ses positions sont interprétées en termes de descripteurs (ou attributs) de natures diverses dans l'espace de description, et une cartographie sémantique entre ces deux espaces est nécessaire. Elle est assurée par des mécanismes d'interprétation qui font correspondre les valeurs continues dans le premier espace à des valeurs appropriées dans le second. Selon l'attribut qui est pris en compte dans l'espace de description, l'un des mécanismes suivants est utilisé pour l'interprétation des positions de l'espace des particules :



**Figure 3.7 Mécanismes d'interprétation entre l'espace de recherche et l'espace de description.**

- L'identité ou la transformation affine pour les attributs continus. Ainsi, la particule à coordonnées continues correspond à une valeur continue dans

l'espace de description comme c'est le cas dans la PSO standard.

- L'arrondissement des valeurs d'attributs entiers, comme il est d'usage de le faire en PSO discrète (voir (Omran, Engelbrecht et Salman, 2007)). Cette méthode arrondit la valeur continue de la particule à l'entier le plus proche pour générer la solution discrète dans l'espace de description.

- La substitution de fréquence pour les attributs catégoriels. La valeur d'une position de particule dans l'espace de recherche correspondant à un descripteur catégoriel est interprétée comme une fréquence du descripteur dans l'espace de description. Un tableau de fréquences est construit au cours d'une étape de prétraitement pour mettre en œuvre la cartographie. Il est construit par l'analyse de la base de cas pour calculer la fréquence de chaque valeur de descripteur (voir tableau 3.3 pour un exemple). Le tableau est utilisé pour interpréter les positions des particules correspondant aux descripteurs catégoriels.

Grâce aux mécanismes d'interprétation, l'algorithme PSO fonctionne toujours en mode continu, mais l'interprétation des positions de particules change. Ce changement a lieu avant l'évaluation de la fonction d'adaptation qui exprime toujours la sémantique de l'espace de description..

Un problème se pose dans le cas où un descripteur a des valeurs ayant de fréquences identiques<sup>11</sup>. Notre solution est de sélectionner l'une des deux fréquences au hasard lors de l'interprétation, ce qui reste conforme à l'esprit d'exploration stochastique de l'espace de recherche.

Donc, la vitesse, la position et l'inertie évoluent toujours dans un espace de recherche continu, mais la fonction d'adaptation  $\psi$  est évaluée avec les valeurs

---

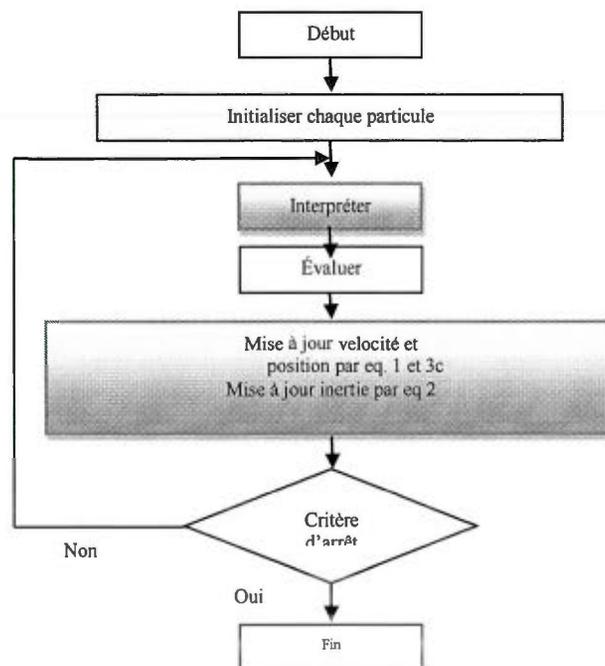
<sup>11</sup> Dans l'ensemble des expérimentations menées, le cas des fréquences identiques ne c'est pas présenté. Néanmoins, nous avons réfléchi à la solution.

interprétées de descripteurs mixtes dans l'espace de description. En d'autres termes, on cherche dans l'espace de recherche, mais on compare dans l'espace de description.

**Tableau 3.2** Tableau partiel des Fréquences pour deux descripteurs de la base de données 'Adult'.

Descripteur	Valeur	Fréquence
Race	'Black'	0.0955
	'Amer-Indian-Eskimo'	0.0090
	'Asian-Pac-Islander'	0.0310
	'White'	0.8567
	'Other'	0.0077
Sex	'Male'	0.6706
	'Female'	0.3294

Le nouvel algorithme PSO sera comme suit (les boîtiers grisés dans la figure 3.8 indiquent les changements effectués sur l'algorithme original):



**Figure 3.8** Organigramme de la PSO à valeurs mixtes.

En termes de pseudo-code :

---

**Algorithme PSO Mixte-Améliorée**

---

```

(1)  début
(2)  pour chaque particule i
(3)      initialiser position et vitesse
(4)  fin-pour
(5)  Tant que (non nombre maximum d'itérations atteint)
(6)      pour chaque particule i
(7)          interpréter la position (*)
(8)          calculer la fonction objective  $\psi_i(t)$  avec les valeurs interprétées
(*)
(9)          Si  $\psi_i(t)$  meilleure que  $\psi(b_i(t))$ 
(10)             alors  $\psi(b_i(t)) = \psi_i$  et  $b_i(t) =$  position courante
(11)          Si  $\psi_i(t)$  meilleure que  $\psi(b_g(t))$ 
(12)             Alors  $\psi(b_g(t)) = \psi_i(t)$  et  $b_g(t) =$  position courante
(13)      fin-pour
(14)      pour chaque particule i
(15)          calculer la vitesse selon équation (1)
(16)          mettre à jour la position de la particule selon l'équation (3c)
(*)
(17)      fin-pour
(18)      mettre à jour le facteur d'inertie selon l'équation (2)
(19)  fin-tant que
(20)  fin

```

---

(\*) Changements apportés à l'algorithme original

Comme mentionné plus haut (section 5.1), la fonction d'adaptation  $\psi$  dépend du contexte d'application. Pour la recherche de cas guidée par la réutilisabilité, on peut l'exprimer en termes du coût de transfert discuté à la section 3:

$$\psi(s, c) = \text{Similarité}(s, c) * \text{Coût\_Copie}(s, c) + \text{Dissimilarité}(s, c) * \text{Coût\_Adaptation}(s, c) \quad (5)$$

où  $s$  et  $c$  sont les problèmes source et cible, c.-à-d. la partie description du problème issue du cas source trouvé et la requête introduite par l'utilisateur.

Notons que l'ajout du mécanisme d'interprétation augmente l'aptitude de la PSO à traiter des données mixtes de manière uniforme sans augmenter la complexité algorithmique de l'approche comme nous pourrons le voir dans le chapitre VI.

### 3.6 Conclusion

La mémoire de cas idéale est celle qui simultanément augmente la vitesse du rappel et améliore la précision et la robustesse de la tâche accomplie par le raisonneur, en particulier l'étape de réutilisation (Bichindaritz, 2008). Dans cette thèse, nous avons ajouté la remémoration guidée par la réutilisation (RGR) à la fonction objective de manière que la fonction à optimiser par PSO exprime une sémantique de réutilisation.

Notre approche s'appuie sur le principe de mémoire accessible par le contenu comme les CRNs. En outre, elle évite l'inconvénient des approches classiques basées index. Elle évite également la nécessité du calcul préalable et la construction de liens de similarité qui ajoutent à la complexité de la construction de modèles de mémoire comme c'est le cas pour Fish & Shrink et les CRNs.

L'approche représente un cadre général. Cependant, lors de l'examen d'un champ d'application spécifique, un réglage des paramètres est nécessaire afin d'améliorer la convergence. Nous discuterons ce sujet plus en détail au chapitre suivant pour la mise en place du cadre applicatif et au chapitre 6 pour le calibrage de l'approche en fonction du contexte d'application.

## CHAPITRE IV

# Cadre applicatif et Modélisation des connaissances

1. Introduction.
2. Ancrage méthodologique et justification du choix.
3. Cadre applicatif.
4. Architecture générale.
5. Modèle des connaissances pour le CBP.
6. Conclusion



#### 4.1 Introduction

L'idée sous-jacente au CBR est de tirer profit des solutions de problèmes déjà résolus pour trouver celle d'un nouveau problème. Par exemple, les Systèmes de Transport Intelligents (STI) utilisent intensivement des informations en temps-réel sur le réseau routier. On peut penser alors qu'en traitant ces informations dans un cadre de réutilisation CBR, on peut prédire rapidement des solutions aux problèmes de congestion du moment et les communiquer aux intéressés, améliorant ainsi le rendement des infrastructures de transport. Le cadre applicatif de notre travail s'inscrit dans cette perspective.

Ce chapitre décrit une analyse des STI comme domaine d'application avec la mise en exergue des besoins utilisateurs. Tel que mentionné dans l'introduction de cette thèse, l'objectif n'est pas l'exhaustivité, mais la représentativité. Aussi, nous aborderons différents aspects à des niveaux de détails différents selon le besoin. Dans la suite, nous présenterons la modélisation de la planification d'itinéraire personnalisé en présentant l'architecture globale ainsi que la représentation des connaissances. Mais auparavant, nous nous intéressons à la justification du choix de l'outil méthodologique puis à la mise en contexte.

#### 4.2 Ancrage méthodologique et justification du choix

Dans le chapitre I, nous avons mis en exergue les caractéristiques du CBR. Dans cette section, nous revenons sur une caractéristique essentielle dans un contexte applicatif, la résolution paresseuse de problèmes. Cette caractéristique permet de mettre le CBR en avant de la scène au moment de choisir une approche de résolution de problème. Parmi les multiples avantages des approches paresseuses (Aha, 1998) et qui s'appliquent tous au CBR, on retrouve :

- L'élicitation : les approches paresseuses nécessitent la disponibilité de cas plutôt que des règles difficiles à extraire. Cela peut considérablement recentrer les efforts de l'acquisition des connaissances sur la façon de structurer les cas.
- Le biais de résolution de problèmes: parce que les cas sont sous forme brute, ils peuvent être utilisés pour différentes fins de résolution de problèmes. En revanche, les règles et autres abstractions ne peuvent généralement être utilisées que pour le but qui a guidé leur compilation.
- L'apprentissage incrémental: les approches paresseuses sont généralement à faible coût d'apprentissage (c'est-à-dire: le traitement des données) en comparaison avec les approches qui tentent de compiler des données dans des abstractions concises. Toutefois, les approches paresseuses requièrent davantage de travail pour répondre aux requêtes.
- Espaces de solution disjonctifs : les approches paresseuses sont souvent plus appropriées pour des tâches dont l'espace des solutions est complexe.
- Explications précédentes : Du fait de stocker des cas plutôt que d'écarter les données, il y a donc une meilleure traçabilité du raisonnement de résolution. Les approches paresseuses peuvent générer des explications précédentes (qui sont basées sur la recherche des cas). Des explications caractéristiques (sous forme d'abstractions) peuvent toujours être tirées des cas stockés, le cas échéant.
- Résolution séquentielle: les tâches séquentielles bénéficient de la conservation d'un historique sous la forme d'états qui conduisent à la situation actuelle. Les approches paresseuses sont utilisées pour stocker ces informations, qui peuvent ensuite être utilisées.

Toutefois, la raison majeure pour l'utilisation des approches paresseuses de résolution de problèmes est qu'elles sont très intuitives ; les experts ont souvent tendance à lier leur comportement de résolution de problème de manière à suggérer une forme de raisonnement basé cas. Au-delà de l'intuition, un autre argument important présenté par Hammond, cité dans (Brown, 1994) est que :

*“Every time you run into a problem which in standard AI technology would be solved by searching a space of possibilities, in CBR what you do is search the space of actuality; you search the space of what does happen, what has happened, rather than the space of what could happen; because the space of what could happen is outrageously huge, and the space of what has happened is nice and tiny and tractable in comparison”.*

La construction automatique d'itinéraires de qualité supérieure qui répondent aux préférences du conducteur (planification personnalisée) constitue un domaine d'application idéal pour le CBP.

Dans la suite de ce chapitre, nous présentons notre approche de planification d'itinéraire personnalisé en focalisant sur l'étape de remémoration.

#### 4.3 Cadre applicatif

Dans le cadre des STI, deux ensembles de marché<sup>12</sup> nous intéressent essentiellement :

- Le guidage routier par FSI (Fournisseurs de Services d'Information) : Cet ensemble de marché offre aux utilisateurs des fonctions évoluées de planification de trajet et de guidage routier adapté aux conditions courantes. Il est différent de

---

<sup>12</sup> Un ensemble de marché est une unité qui donne une perspective accessible de l'architecture, axée sur le déploiement. Les ensembles de marché sont adaptés - séparément ou selon diverses combinaisons - à des besoins et à des problèmes réels de transport. Ils identifient les éléments de l'architecture physique qui sont requis pour la mise en œuvre d'un service de transport particulier (source : site web des systèmes de transport intelligents du ministère des transports du Canada).

l'ensemble Guidage routier dynamique<sup>13</sup>, car il transfère la fonction de planification du trajet effectuée par le dispositif de l'utilisateur au fournisseur de services d'information (FSI). Cette méthode simplifie le matériel de l'utilisateur et peut fournir à l'infrastructure de meilleures informations pour prévoir le trafic et des stratégies de contrôle appropriées pour prendre en charge la planification de base des trajets avec un minimum de matériel de l'utilisateur. Cet ensemble offre à la fois les fonctions de guidage routier systématique comme celles qui peuvent être utilisées dans un véhicule et les fonctions d'établissement des trajets avant le voyage. Cet ensemble permet la transmission bidirectionnelle des données, et facultativement, le véhicule est doté des bases de données, des capacités de localisation et de la technologie d'affichage nécessaires pour la prise en charge du guidage routier systématique.

- Diffusion de l'information sur le trafic : Cet ensemble de marché permet la diffusion d'information sur le trafic aux conducteurs et aux véhicules en utilisant du matériel en bordure de route comme les panneaux à message dynamique ou le service radio d'avis routiers. Il fournit un outil pouvant servir à avertir les conducteurs des incidents. Le matériel, placé à des endroits stratégiques en bordure de route, permet de diffuser l'information à des endroits où les conducteurs peuvent réagir et adapter leur itinéraire en conséquence. Cet ensemble de marché couvre aussi le matériel et les interfaces qui transmettent de l'information sur le trafic d'un centre de gestion du trafic aux médias (par exemple, au moyen d'un raccordement direct entre les systèmes informatiques d'un centre de gestion du trafic et d'une station radio ou de télévision), à un centre de gestion du transport en commun, à un centre de gestion des urgences et à un fournisseur de services d'information.

---

<sup>13</sup> Le guidage routier dynamique est un ensemble de marché qui offre aux utilisateurs des fonctions évoluées de planification de trajet et de guidage routier adapté aux conditions courantes. Cet ensemble combine le matériel de guidage routier autonome de l'utilisateur à un récepteur numérique pouvant recevoir des données en temps réel sur le trafic, le transport en commun et l'état des routes, dont le matériel de l'utilisateur tient compte pour effectuer le guidage routier (source : site web des systèmes de transport intelligents du ministère des transports du Canada).

Ces deux ensembles de marché constituent le cadre spécifique dans lequel le travail de recommandation de chemins alternatif va s'insérer. En gros, l'idée est d'augmenter l'offre de service faite au niveau de ses ensembles. De deux manières : d'une part ajouter au système une aptitude de guidage en fournissant un chemin alternatif et augmenter ces capacités de calculs au moyen d'une technique d'optimisation par essais de particules.

Ce sous-contexte applicatif reste de complexité relativement importante. Aussi, nous allons limiter notre exposé dans la partie modélisation de sorte à illustrer le contexte et non le reproduire exhaustivement.

#### 4.3.1 Modélisation

À partir de ces deux ensembles de marché se dégagent les acteurs principaux dont les buts vont déterminer les cas d'utilisation exprimant les besoins à modéliser :

Acteurs	Buts
Conducteur	<ul style="list-style-type: none"> <li>• Circuler d'une origine vers une destination.</li> <li>• Demander un chemin alternatif.</li> <li>• Suivre une recommandation.</li> <li>• Respecter signalisation.</li> <li>• Communiquer.</li> </ul>
Contrôleur	<ul style="list-style-type: none"> <li>• Réguler trafic.</li> <li>• Guider véhicule.</li> <li>• Donner recommandation.</li> <li>• Demander calcul.</li> <li>• Communiquer.</li> </ul>
Intersection	<ul style="list-style-type: none"> <li>• Afficher recommandation.</li> <li>• Collecter les données.</li> </ul>
Calculateur	<ul style="list-style-type: none"> <li>• Calculer chemin.</li> <li>• Calculer segment de chemin.</li> <li>• Demander calcul de segment de chemin à une autre zone.</li> <li>• Intégrer différents segments.</li> </ul>
Informateur	<ul style="list-style-type: none"> <li>• Fournir diverses données sur le trafic.</li> <li>• Mise à jour des données.</li> </ul>

#### 4.3.1.1 Spécification des Acteurs

##### ***Calculateur***

Il s'agit en fait du sous-système qui procèdera au calcul du chemin alternatif, au moyen de la technique PSO. Du point de vue de la recommandation, il est vu comme un acteur.

##### ***Conducteur***

C'est l'un des acteurs essentiels dans la mesure où l'offre de service s'adresse à lui. Dans le sous-contexte de résolution, il représente aussi le véhicule.

##### ***Contrôleur***

C'est une entité qui supervise l'ensemble des opérations de gestion du trafic. Dans ce sous-contexte, on ne voit que les tâches qui touchent de manière directe les ensembles de marché qui nous intéressent.

##### ***Informateur***

Là aussi il s'agit d'un sous-système qui régit l'ensemble des données disponibles sur : le trafic routier, les données météorologiques, les données sur les incidents ainsi que les travaux du réseau routier (entretien, maintenance et déneigement). En fait, il y a un informateur par zone géographique. L'ensemble des informateurs constitue un entrepôt de données complexe. Un informateur est vue comme un acteur dans la mesure où il gère des données (fournit et maintien à jour). L'informateur inclut toute la connaissance sur le trafic (ce qui nous dispense de représenter le trafic explicitement).

##### ***Intersection***

Cette entité physique a été représentée sous forme d'acteur du domaine puisque c'est au niveau d'une intersection que l'information est collectée ou transmise au conducteur.

#### 4.3.1.2 Spécification des cas d'utilisation

Description graphique.

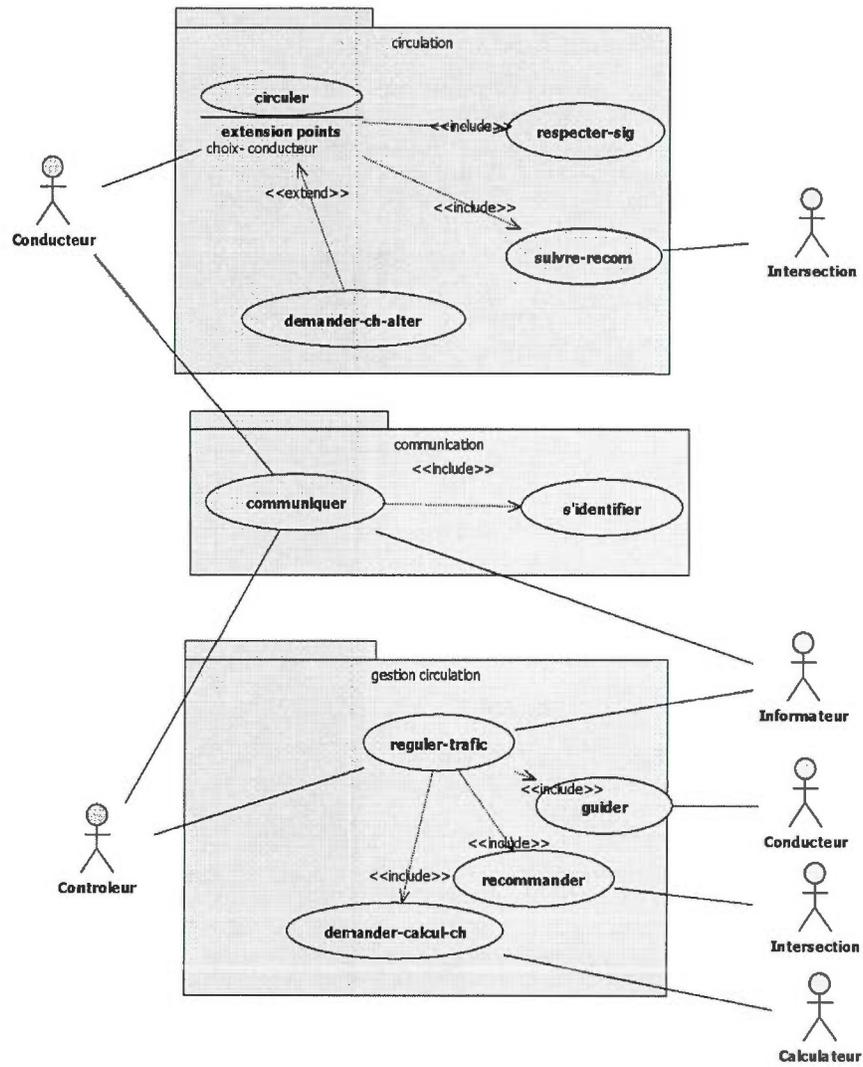


Figure 4.1 Cas d'utilisation global

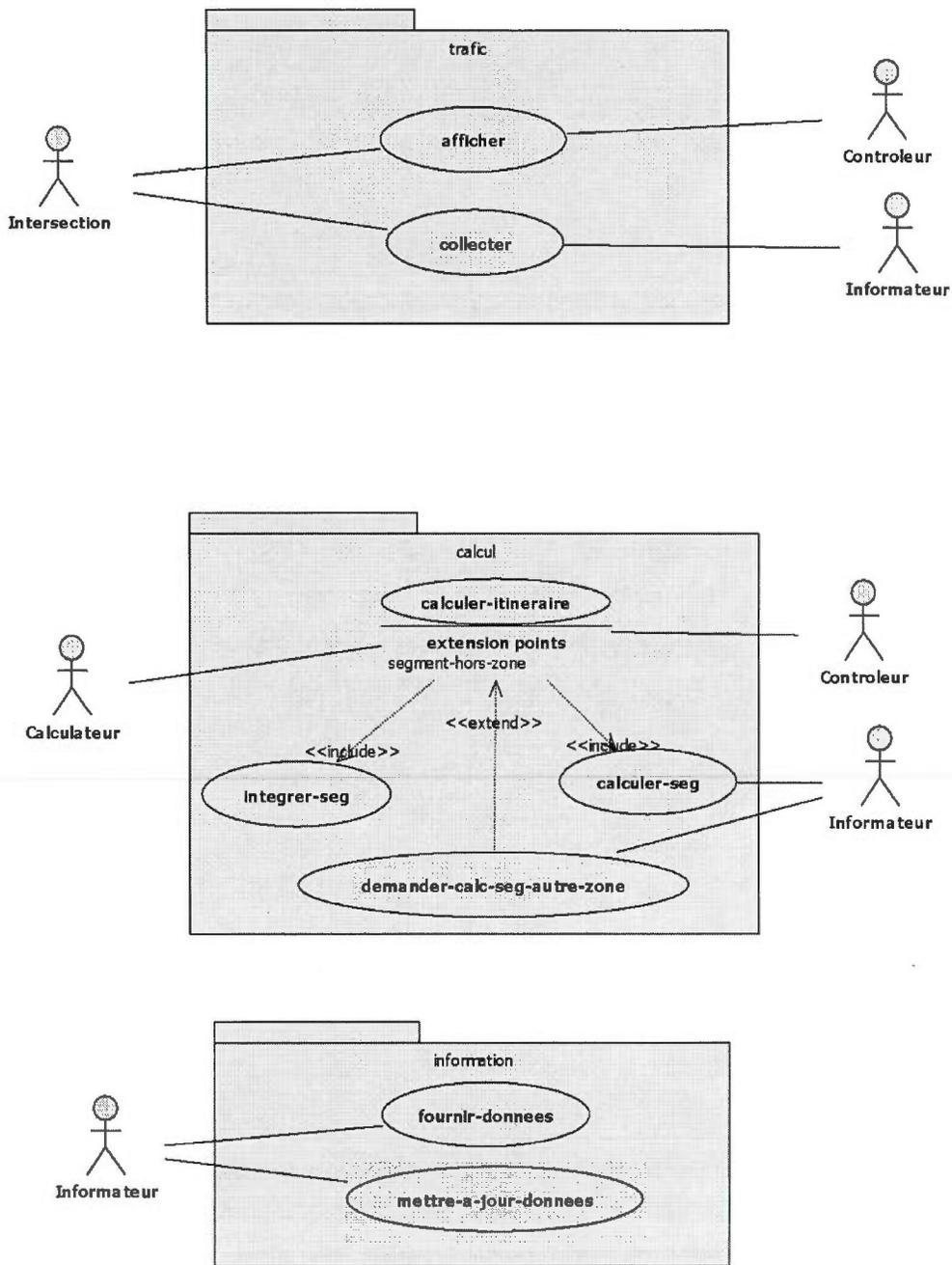


Figure 4.2 Les cas d'utilisation organisés en package

## Description Textuelle.

À titre illustratif, nous donnons une description textuelle de quelques cas d'utilisation:

<p>Titre : Afficher          But : Permet l'affichage du chemin alternatif recommandé.          Acteurs : Intersection, Contrôleur.          Contraintes :</p>
<p>Pré-conditions : Flux normal ralenti et itinéraire alternatif fourni par calculateur.</p>
<p>Enchaînements : Affiche un message recommandant un itinéraire alternatif sur un panneau situé à l'approche d'une intersection critique.</p>
<p>Post-condition : Recommandation affichée.</p>
<p>Titre : Calculer-itinéraire          But : Procède à la première étape de calcul du chemin alternatif en segmentant le chemin.          Acteurs : Calculateur, Contrôleur, Informateur          Contraintes : Réponse en temps réel.</p>
<p>Pré-conditions : Requête de calcul faite par le contrôleur de la même zone ou provenant du calculateur d'une autre zone.</p>
<p>Enchaînements :          A partir de l'information fournie par l'informateur sur la structure du chemin, il y a segmentation du chemin.          Lorsque le point destination se trouve en dehors de la zone géographique à laquelle est associé le calculateur, on lance une requête de calcul vers le calculateur de la zone appropriée.</p>
<p>Post-condition : Fourniture d'une fragmentation du chemin en segments de chemin.</p>
<p>Titre : calculer-seg.          But : Il s'agit du calcul d'un fragment de chemin optimal.          Acteurs : Calculateur, Contrôleur, Informateur          Contraintes : Réponse en temps réel.</p>
<p>Pré-conditions : Segmentation.</p>
<p>Enchaînements :          À partir des données fournies par l'informateur sur l'état du trafic dans le segment, il y a calcul du meilleur segment à emprunter.          Le calcul en soit se fait selon une approche que nous allons détailler dans le chapitre suivant.          Lorsque le point destination se trouve en dehors de la zone géographique à laquelle est associé le calculateur, on lance une requête de calcul vers le calculateur de la zone appropriée.</p>
<p>Post-condition : Fourniture des meilleurs segments pour la zone en cours.</p>

<p><b>Titre :</b> Circuler</p> <p><b>But :</b> Permet au conducteur de partir d'un point d'origine vers un point de destination.</p> <p><b>Acteurs :</b> Conducteur, Intersection</p> <p><b>Contraintes :</b></p>
<p><b>Pré-conditions :</b> Disposer d'un véhicule et d'un itinéraire.</p>
<p><b>Enchaînements :</b></p> <p>En temps normal et lorsque la circulation est fluide, le conducteur suit un itinéraire qu'il s'est tracé d'avance et ce en respectant la signalisation et éventuellement en suivant les recommandations d'usage.</p> <p>La signalisation peut être sous forme de feu tricolore, panneaux indicatifs figés ou changeants.</p> <p>En cas de ralentissement du flux ou arrêt, le conducteur peut décider d'emprunter un chemin alternatif.</p>
<p><b>Post-condition :</b> Atteindre la destination.</p>
<p><b>Titre :</b> Collecter</p> <p><b>But :</b> Il s'agit de la collecte d'information de manière très localisée, au niveau d'une intersection.</p> <p><b>Acteurs :</b> Intersection</p> <p><b>Contraintes :</b> d'ordre logistique</p>
<p><b>Pré-conditions :</b> Disposer d'équipement de capture et de stockage temporaire.</p>
<p><b>Enchaînements :</b></p> <p>Les données peuvent provenir de divers dispositifs de collecte : caméra vidéo, capteurs en bordure de route, dispositif de comptage... Tous situés à proximité d'une intersection. Les données seront collectées puis acheminées via un réseau de communication vers l'informateur.</p> <p>En cas de défaillance d'un dispositif de capture quelconque, la défaillance est signalée au contrôleur de la zone.</p>
<p><b>Post-condition :</b> Fournir données locales sur trafic.</p>
<p><b>Titre :</b> Communiquer</p> <p><b>But :</b> Il s'agit de communiquer l'information via le réseau de communication.</p> <p><b>Acteurs :</b> Conducteur, Contrôleur, Informateur.</p> <p><b>Contraintes :</b></p>
<p><b>Pré-conditions :</b> Disposer d'équipement de communication.</p>
<p><b>Enchaînements :</b></p> <p>Le conducteur éprouve le besoin de communiquer une requête de chemin alternatif via un dispositif GPS ou un téléphone mobile. Sa requête inclura sa position actuelle, sa destination et éventuellement ses préférences.</p> <p>Le contrôleur a des besoins de communication plus importants. Cela peut être :</p> <ul style="list-style-type: none"> <li>• Des données : de diverses natures et provenant de l'informateur.</li> <li>• Des recommandations : ce sont des messages générés suite à un calcul de chemin alternatif et transmis : à l'intersection concernée.</li> <li>• Des directives : ce sont des messages faisant partie du flux décisionnel et peuvent être transmis à n'importe quelle partie du système.</li> </ul> <p>L'informateur peut communiquer de l'information provenant de n'importe quelle origine et la transmettre vers n'importe quelle destination.</p>
<p><b>Post-condition :</b> Message passé.</p>

Titre : Demander-calc-seg-autre-zone But : Il s'agit de demander le calcul pour un segment se trouvant hors zone. Acteurs : Calculateur, Informateur. Contraintes :
Pré-conditions : avoir une partie de l'itinéraire hors zone.
Enchainements : Le calculateur procède à cette demande lorsque le calcul en cours nécessite les données sur une portion de chemin se trouvant hors de portée de l'informateur en cours. En fonction de l'information sur le voisinage fournie par l'informateur en cours, le calculateur formule sa requête et l'adresse à la zone concernée.
Post-condition : Segment calculé.

#### 4.3.2 Diagramme de classes

Dans ce qui suit, nous donnons la liste des classes à considérer dans un premier niveau d'analyse.

Classes	Packages
Calculateur	::Calcul
Communication	::Communication
Conducteur	::Circulation
Controleur	::Gestion circulation
Dem-guidage	::Circulation
Destination	::circulation
Dispo-com	::Communication
GPS	::Communication
Guidage	::Gestion circulation
Info-pertinente	::Information
Informateur	:: Information
Intersection	::Trafic
Itineraire	:: Calcul
Paneau	::Trafic
Recommandation	:: Gestion circulation
Route	::Trafic
Segment	::Trafic
Tel-mob	::Communication
Vehicule	::Trafic
Localization	:: Circulation



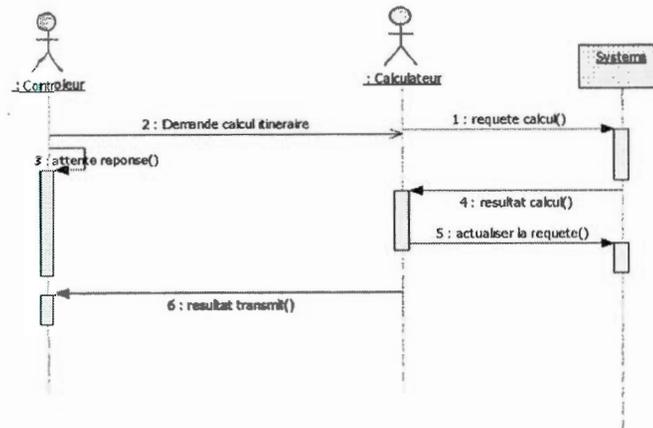


Figure 4.4 Diagramme de séquence1.

Le second diagramme (Figure 4.5) retrace un scénario de demande de calcul pour un segment hors de la zone en cours.

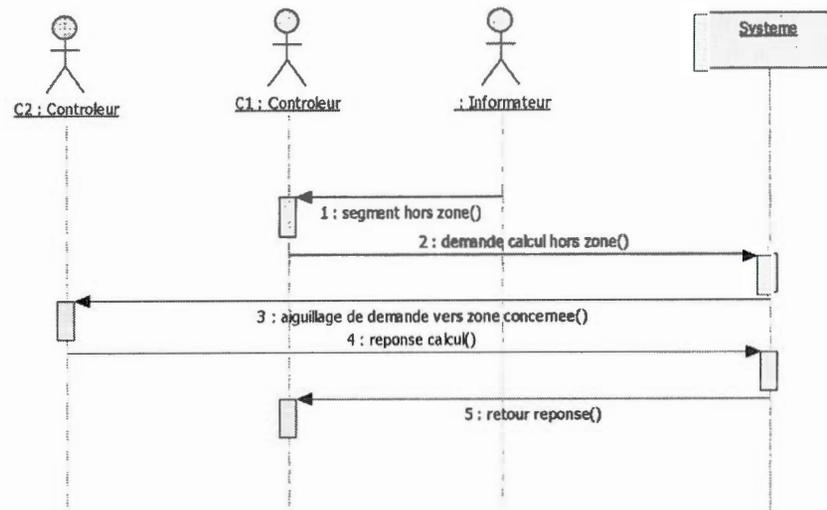


Figure 4.5 Diagramme de séquence2.

#### 4.3.4 Diagramme état/transition.

Comme exemples illustratifs de diagramme état/transition, nous avons choisi les états que peut prendre une requête de calcul de chemin (Figure 4.6). Elle peut émaner du contrôleur et est donc dans son état 'req\_recommandation' ou du conducteur et est sous l'état

'Demand\_chemin'. Ces deux états sont uniformisés en un état 'Req\_calcul' qui boucle sur lui-même dans le cas où une requête de calcul est envoyée vers une autre zone.

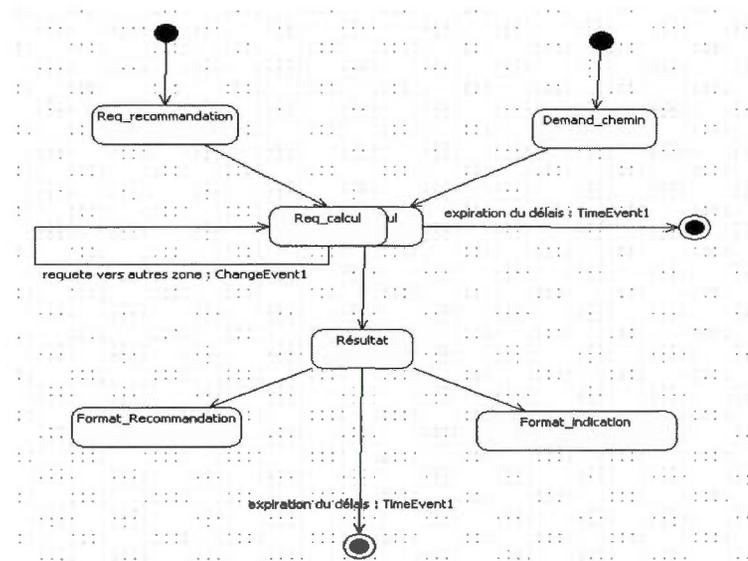


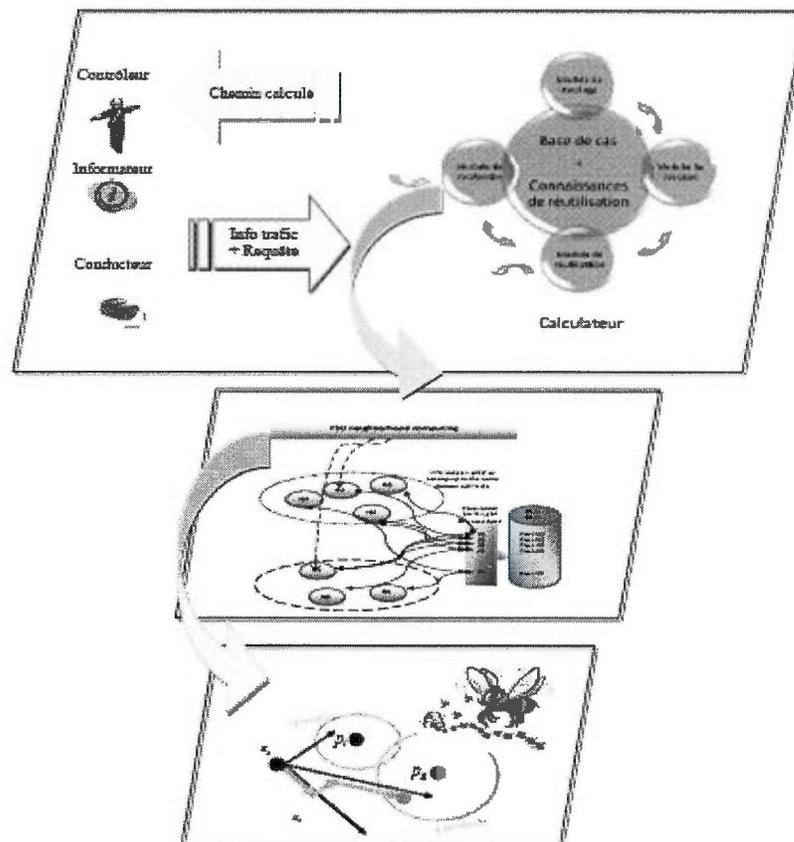
Figure 4.6 Diagramme d'état/transition.

L'état 'Req\_calcul' peut transiter vers un état final terminaison si le délai de calcul a expiré ou encore passer vers un état 'Résultat'. Ce dernier transite vers les états finaux 'Format\_Recommandation' ou 'Format\_Indication' selon la présentation du résultat voulue. Comme il peut transiter vers un état final terminaison si le délai de calcul a expiré.

#### 4.4 Architecture générale

Tout en se référant aux acteurs du domaine tel que décrit précédemment, nous allons mettre le focus sur le sous-système de calcul d'itinéraire appelé *Calculateur* décrit dans la section précédente. Parmi les acteurs du domaine, trois seront considérés comme acteurs externes pour l'architecture en cours (figure 4.7). Il s'agit en l'occurrence de *Conducteur*, de *contrôleur* et de *Informateur*. Les autres acteurs ayant un rôle transparent par rapport au niveau d'analyse en cours. Par ailleurs, précisons que le contexte actuel se rapporte à une zone géographique limitée et que

c'est le niveau macroscopique qui prendra en charge les requêtes portant sur d'autres zones. Notons que l'inférence reste la même pour tous les *calculateurs*.



**Figure 4.7 . Architecture générale.**

Le conducteur fournit une requête mentionnant sa destination, et ses préférences. L'informateur va fournir la position GPS actuelle ainsi que les informations sur le trafic. Le calculateur fournira au contrôleur l'itinéraire calculé.

Notons que la discontinuité dans les flèches symbolisant le flux en entrée et en sortie indique que les données sont traitées et mises sous une forme directement

utilisable par le calculateur afin d'assurer une homogénéité de la représentation. Par exemple une adresse civique sera transformée en coordonnées GPS.

Exemple : Coin Shebrooke / Jeanne Mance sera traduit par le couple  
(45.509374,-73.570851) en coordonnées GPS

Le calculateur quant à lui, est constitué de quatre modules principaux fonctionnant autour d'une base de cas accompagnée des connaissances de réutilisation que nous expliciterons par la suite. Le module de recherche constitue le noyau de notre travail. Son rôle sera de trouver des itinéraires stockés dans la base de cas qui soient réutilisables dans le contexte en entrée. Le module de réutilisation a pour rôle d'apporter les ajustements nécessaires à l'itinéraire remémoré pour répondre aux exigences de la requête. Le module de révision procède à l'évaluation de l'itinéraire ainsi généré afin de décider de le stocker pour une utilisation future ou non. Lorsque le nouvel itinéraire aura passé la phase d'évaluation, son stockage en mémoire de cas sera pris en charge par le module de stockage.

Dans la suite, nous focaliserons notre effort d'investigation sur le module de remémoration. Pour ce faire, présentons d'abord la modélisation des connaissances pour le CBP.

#### 4.5 Modélisation des connaissances pour le CBP

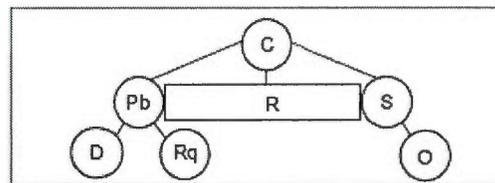
Dans ce qui suit, nous précisons la modélisation d'un cas appliquée à la tâche de planification. Tout d'abord rappelons qu'un cas noté  $C$  est une connaissance spécifique représentant un épisode de résolution de problème composé de trois parties principales (voir figure 4.8) : le problème  $P_b$ , sa solution  $S$  et le raisonnement  $R$ .

Le problème est décrit par un ensemble de données descriptives  $D$ , et d'une requête  $R_q$  indiquant l'objectif du raisonnement, toujours connu. Les données du

problème comportent des contraintes à satisfaire par la solution, et constituant une spécification partielle de S.

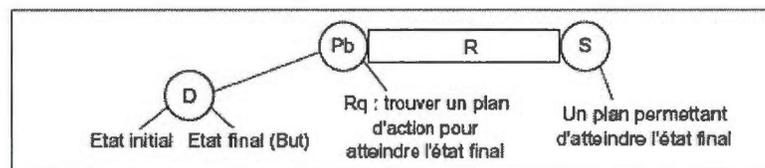
La solution S construite par R doit permettre de satisfaire les objectifs spécifiés par Pb. La solution à construire par R peut être un objet quelconque (un plan, dans ce cas).

Le raisonnement R est l'ensemble des étapes de raisonnement, les résultats intermédiaires, les décisions prises pour passer de Pb à S. Les connaissances représentées dans les cas sont aussi bien descriptives (Pb,S), qu'opérationnelles ( R ).



**Figure 4.8** Un cas C est composé du problème Pb, du raisonnement R et de la solution S (extraite de (Fuch, 1997))

L'application de cette définition du cas à la tâche de résolution 'planification' (figure 4.9), permet d'obtenir la définition suivante (Fuch, 1997) : « Le problème comprend le descriptif d'un état initial et de l'état final à atteindre par la solution. Le raisonnement construit un plan permettant d'atteindre l'état final à partir de l'état initial ».



**Figure 4.9** Un cas C pour la Planification (extraite de (Fuch, 1997)).

Dans le cadre de la planification de chemin, une description de cas serait le n-uplet :

Cas = <Origine, Destination, Chemin, Préférences >

Origine : est la position courante en longitude et latitude.

Destination : est le but à atteindre exprimé dans ce cas aussi en longitude et latitude.

Chemin : séquence de segments menant de l'origine à la destination.

Préférences : ensemble de choix de l'utilisateur concernant l'itinéraire pris. Les préférences constituent une partie du contexte dans lequel l'itinéraire a été suivi.

Un segment : est une portion de chemin délimité par un point de départ et un point d'arrivée (chacun de ces points constitue en fait une intersection avec un autre segment), ayant une longueur et une forme connues ainsi que la vitesse autorisée. Ces deux dernières informations donneraient une indication de la durée du parcours du segment.

Techniquement parlant, on disposerait des structures suivantes :

**Tableau 4.1 Description du cas**

Dimension	Description
Origine	(Longitude, Latitude)
Destination	(Longitude, Latitude)
Chemin	{Segment}
Préférences	{préférence}

**Tableau 4.2 Description d'un segment**

Sous dimension	Description
Début	(Longitude, Latitude)
Fin	(Longitude, Latitude)
Sens	{unique, double}
Longueur	Numérique
Forme	{droit, virage_aigu, virage_obtu, pente, descente}
Vitesse autorisée	Numérique ([30,110])

**Tableau 4.3 Description d'une préférence**

Sous-dimension	Description
Valeur	{nulle, autoroute, route_ville, route_campagne, route_habituelle, sans_pente, sans_virage}
Instant_début	Numérique (nul lorsque non utilisé, négatif lorsqu'il faut éviter)
Instant_fin	Numérique (nul lorsque non utilisé, négatif lorsqu'il faut éviter)

Du point de vue mémoire de cas, les nœuds-valeurs vont prendre leurs valeurs dans les domaines de définition des descripteurs ou dimensions suivants :

**Tableau 4.4 Aperçu des valeurs possible des EIs**

Dimension	Description	Valeurs possibles
Origine	Symbole associé au couple (Longitude, Latitude)	A1, A2, ...
Destination	Symbole associé au couple (Longitude, Latitude)	A1, A2, ...
Chemin	Segment <sub>i</sub> -Début : Symbole associé au couple (Longitude, Latitude)	A1, A2, ...
Chemin	Segment <sub>i</sub> -Fin : Symbole associé au couple (Longitude, Latitude)	A1, A2, ...
Chemin	Segment <sub>i</sub> -Sens	{unique, double}
Chemin	Segment <sub>i</sub> -Longueur	Numérique
Chemin	Segment <sub>i</sub> -Forme	{droit, virage_aigu, virage_obtu, pente, descente}
Chemin	Segment <sub>i</sub> -Vitesse-autorisée : Limite supérieure de la vitesse pour segment <i>i</i>	{30, 40,50, 60, 70,100,110}
Chemin	...	
Chemin		
Chemin	Segment <sub>n</sub> -Vitesse-autorisée : Limite supérieure de la vitesse pour segment <i>n</i>	{30, 40,50, 60, 70,100,110}
Préférences	Préférence <sub>i</sub> -Valeur	{nulle, autoroute, route_ville, route_campagne, route_habituelle, sans_pente,

		sans virage}
Préférences	Préférence <sub>i</sub> - Instant_début	{nul, négatif,positif}
Préférences	Préférence <sub>i</sub> - Instant_Fin	{nul, négatif,positif}
Préférences	....	
Préférences		
Préférences	Préférence <sub>m</sub> - Instant_Fin	{nul, négatif,positif}

La requête émise par le conducteur serait :

Req = < Destination, Préférences >

**Destination** : est le but à atteindre exprimé en adresse postale, elle subira par la suite une transformation en longitude et latitude pour être directement utilisable par le calculateur.

**Préférences** : ensemble de choix de l'utilisateur concernant l'itinéraire à prendre.

Cette requête sera complétée par des informations sur le contexte actuel provenant de *Informateur* et portant sur :

**Évènements** : informations se rapportant aux incidents, manifestations ou autres pouvant éventuellement ralentir le trafic.

**Travaux** : présence ou absence de travaux (réfection ou déneigement) dans un segment donné.

Les connaissances de réutilisation quant à elles, se présentent sous deux formes :

- Les connaissances de similarité : c'est l'ensemble des métriques et heuristiques utilisées pour évaluer le rapprochement entre deux cas.
- Les connaissances d'adaptation : utilisées lors de l'adaptation d'un ancien plan pour satisfaire la requête du conducteur

Dans le cas de la planification d'itinéraire, la partie problème sur laquelle portera l'étude de similarité est constituée des descripteurs ou dimensions Destination et Préférences.

Les parties de la solution sur lesquelles porteront les modifications sont les segments. À ceux-là s'ajouteront les dimensions du contexte fournies par *Informateur*, à savoir la météo de manière globale et les travaux de manière locale relativement à un segment particulier. Les segments seront soit supprimés (lorsqu'ils sont en trop) soit ajoutés (pour compléter un itinéraire) soit remplacés par d'autres segments en fonction des préférences et du contexte. À cet effet, une liste de segments alternatifs est disponible comme connaissance d'adaptation.

#### 4.6 Conclusion

Les enjeux économiques, sociaux et écologiques font de la gestion du trafic routier un domaine de recherche très fertile, notamment pour ce qui est des Systèmes de Transport Intelligents.

Nous avons tenté à travers ce chapitre de donner une idée générale de la richesse de ce domaine, de sa complexité, mais surtout de montrer un cadre applicatif réel. L'idée étant de circonscrire le sous-contexte dans lequel nous pourrions intervenir sans pour autant en faire une étude exhaustive. Ce chapitre peut être vu comme les balises d'une passerelle de la thèse académique vers le domaine industriel.

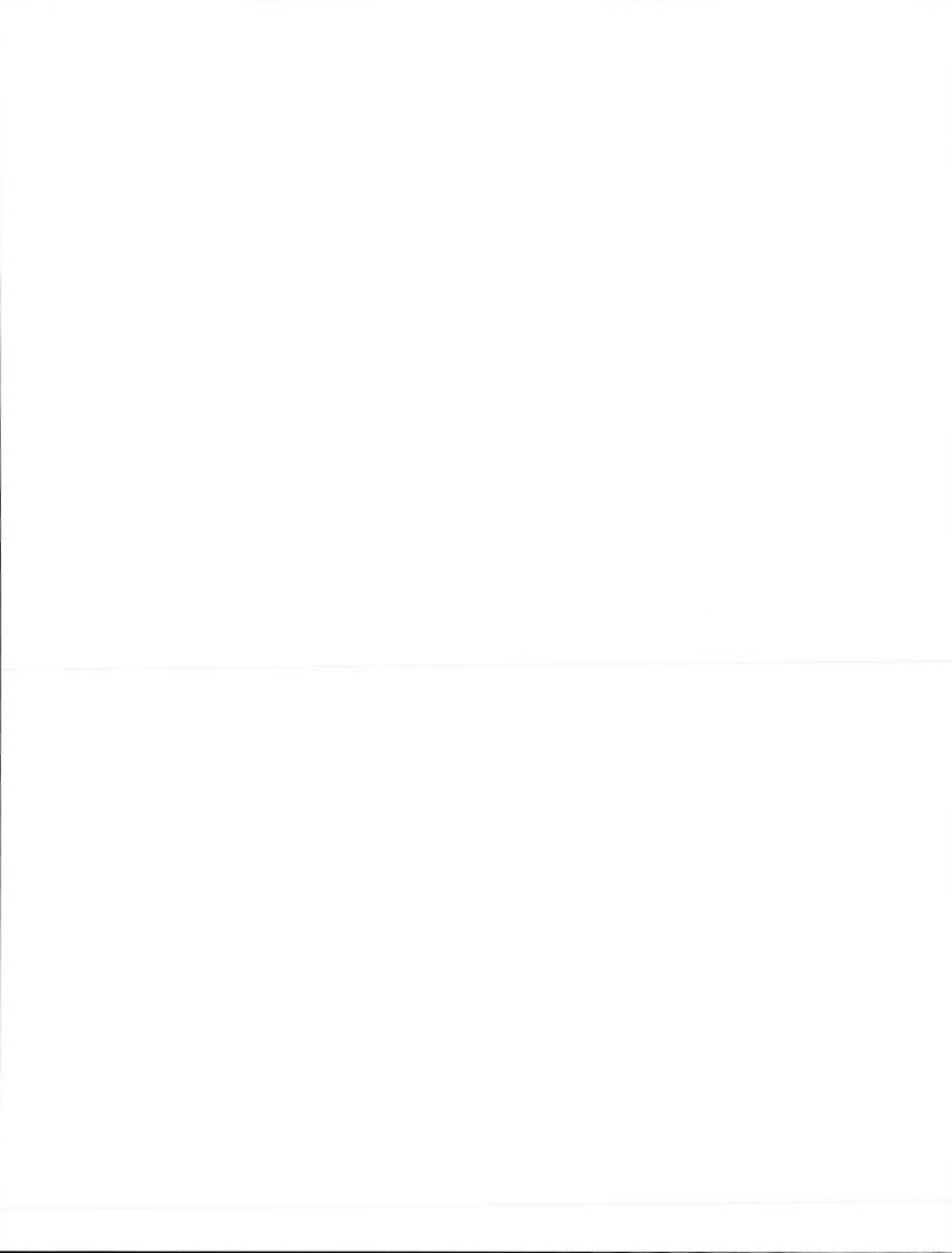
En dépit de la non-exhaustivité du sous-contexte applicatif, la modélisation des connaissances qui en découle permet la mise en place d'une partie de notre protocole expérimental (étape 4 du chapitre 5) et, par cette non-exhaustivité même, nous avons pu motiver les étapes 1, 2 et 3 du protocole expérimental du chapitre 5.



## CHAPITRE V

# Protocole expérimental

1. Introduction
2. Démarche
3. Connaissances utilisées
4. Environnement
5. Conclusion



## 5.1 Introduction

Ce chapitre décrit la démarche prise afin d'expérimenter les choix méthodologiques effectués dans la partie 2. Comme certaines décisions conceptuelles prises dans notre travail sont inédites (c'est le cas pour le traitement des données mixtes) et de portée générique, elles sont validées de manière transversale avec des tâches autres que la planification, en l'occurrence la classification et la conception. Dans ce cadre, nous présentons dans la suite de ce chapitre les étapes d'expérimentation, les ensembles de données utilisés ainsi que l'environnement de programmation.

## 5.2 Démarche.

Nous avons adopté une démarche de résolution incrémentale où chaque étape vise à résoudre un objectif intermédiaire pour le développement de l'étape suivante. Nous présentons la démarche de validation de ces étapes comme suit :

***Étape 1 : « Développement d'un algorithme PSO pour la prise en charge de données mixtes »***

L'objectif premier ici est l'implémentation d'un algorithme PSO capable de fonctionner avec une description de problème comportant des dimensions de nature mixte. Suit alors le test et la validation de l'algorithme avec des données à description mixte. La disponibilité des données et la diversité des travaux dans le domaine de la classification automatique nous ont poussés à mettre au point une approche basée PSO pour classifier les données mixtes dotée d'un mécanisme d'interprétation spécifique. L'approche est testée à des fins de comparaison sur les bases publiques 'Adult', 'CMC' et 'Abalone' présentant des données mixtes et aussi une grande dimensionnalité pour la base 'Adult'. L'algorithme validé est repris comme noyau pour l'étape 2.

***Étape 2 : « un premier modèle de mémoire basé PSO »***

L'objectif de cette étape est d'implémenter un modèle de mémoire de cas qui respecte un critère de recherche basé similarité et qui repose sur une approche par essaim de particules pour la recherche en mémoire. Pour se faire, une fonction objective représentant une sémantique de similarité sera utilisée dans l'algorithme PSO. L'expérimentation de cette étape est mise en œuvre sur un ensemble de données à description mixte et de grande taille, en l'occurrence la base publique 'Adult'. La validation est faite relativement à l'algorithme de recherche séquentielle (algorithme de force brute appliqué à une mémoire plate) qui est la référence pour la recherche en mémoire grâce à sa caractéristique d'exhaustivité.

***Étape 3 : « un modèle de mémoire basé PSO intégrant le critère de réutilisabilité »***

L'objectif de cette étape est de tester la performance de l'algorithme de recherche basé PSO après l'ajout du critère d'adaptabilité. L'algorithme est le même que dans l'étape 2, mais sa capacité de recherche est rehaussée par l'enrichissement de la sémantique de la fonction objective qui intègre maintenant la sémantique RGR au moyen de l'équation de transfert (équation 5 du chapitre 3). L'expérimentation est faite sur une base de cas propriétaire 'GUIDIET'.

***Étape 4 : « une modélisation CBR pour la planification »***

Cette étape met en place les modèles de connaissance pour capturer de l'expertise de génération de plan, incluant le modèle de connaissances de similarité, d'adaptation et d'apprentissage. Tel que mentionné auparavant, la construction de la base a une visée essentiellement illustrative et la qualité de couverture de la base n'est pas considérée. Ainsi, nous avons mis au point une base de cas portant sur une zone précise du centre-ville de Montréal.

### 5.3 Connaissances utilisées

#### 5.3.1 Dans le cadre des étapes 1 et 2 : « Développement d'un algorithme PSO pour la prise en charge de données mixtes » et « Premier modèle de mémoire basé PSO »

Lors de l'étape 1, nous avons utilisé l'approche PSO décrite à travers l'algorithme PSO mixte amélioré de la section 5.2.3 du chapitre III, avec comme fonction objective celle intégrant la sémantique de distance entre un exemplaire de la classe et la position de la particule courante, décrite dans l'équation suivante :

$$\psi_i^c(t) = \frac{1}{D_{Train}} \sum_{k=1}^{D_{Train}} d(y_k^c, x_i^c(t)) \quad (7)$$

Où  $y_k^c$  est le  $k^{ième}$  exemplaire de la classe  $c$  dans l'ensemble d'apprentissage de taille  $D_{Train}$  et  $x_i^c(t)$  est la position courante du centroid de la classe  $c$  déterminé par la  $i^{ième}$  particule <sup>14</sup>, les deux vecteurs étant de dimension  $N$  (l'indice de la dimension ayant été omis pour alléger la notation).

Lors du calcul de distance, chaque composant à  $N$  dimensions est divisé par son rang, et la somme des distances des composants est divisée par  $N$ . ainsi, n'importe quelle distance sera normalisée dans l'intervalle  $[0.0, 1.0]$  et il en sera de même pour la fonction objective

Lors de l'étape 2, nous avons utilisé l'approche PSO décrite à travers l'algorithme PSO mixte amélioré de la section 5.2.3 du chapitre III, avec comme fonction objective celle intégrant la sémantique de distance entre le descripteur le cas source représenté par la requête et le cas cible potentiel qui représente une instance du

---

<sup>14</sup> Cette position est raffinée durant l'étape d'apprentissage (en utilisant l'ensemble d'apprentissage) pour devenir le centroid qui sera utilisé durant l'étape de décision (en utilisant soit la base de test soit une nouvelle base de données).

voisinage calculé. La fonction est décrite dans l'équation 8 représentant la métrique normalisée de Gower présentée dans la section 3.2 du présent chapitre.

Pour l'expérimentation de ces deux étapes, nous avons fait appel à trois ensembles de données issus de UCI Repository (Asuncion et Newman, 2007). Ces données se caractérisent par la présence d'attributs à valeurs mixtes.

### **La base de données 'Adult'**

En plus de sa caractéristique de forte dimensionnalité, les 15 attributs de chacune de ses 48842 instances sont un mélange de valeurs discrètes, continues et catégorielles. L'ensemble de données a été initialement développé pour la prédiction du revenu annuel d'une personne. Il s'agit de l'une des bases les plus utilisées du point de vue de la taille et de la dimensionnalité par les algorithmes de classification connus.

Les 15 descripteurs de données sont comme suit :

- 1) age : continue.
- 2) workclass : Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
- 3) fnlwgt : continue.
- 4) education : Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
- 5) education-num : continue.
- 6) marital-status : Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
- 7) occupation : Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-

- inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
- 8) relationship : Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
- 9) race : White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, black.
- 10) sex : Female, Male.
- 11) capital-gain : continue.
- 12) capital-loss : continue.
- 13) hours-per-week : continue.
- 14) native-country : United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinidad&Tobago, Peru, Hong, Holand-Netherlands.
- 15) Class :  $\leq 50K$  ,  $> 50K$

Une des particularités de cette base de données est la présence de données manquantes : 7% du nombre total d'instances présente de données manquantes pour certains attributs. Pour régler ce problème, la majeure partie des méthodes de classification commence par éliminer toutes les instances avec données manquantes avant de procéder à la classification.

### **La base de données 'CMC'**

La base de données CMC (pour Contraceptive Method Choice) a été initialement développée pour la prédiction du choix de la méthode de contraception pour une

population de femmes. Les données sont extraites de l'enquête nationale de 1987, faite en Indonésie portant sur la prévalence de la contraception. L'ensemble de données a 1473 cas avec 9 entrées (2 continues, 4 catégorielles et 3 binaires). Le problème est de prédire le choix de la méthode de contraception en cours ('no use', 'long-term methods', ou 'short-term methods') d'une femme, en se basant sur des caractéristiques démographiques et socio-économiques (par exemple : âge de l'épouse et sa religion).

Les données se présentent sur 10 descripteurs :

1. Wife's age	(numérique)	
2. Wife's education	(catégoriel)	1=low, 2, 3, 4=high
3. Husband's education	(catégoriel)	1=low, 2, 3, 4=high
4. Number of children ever born	(numérique)	
5. Wife's religion	(binaire)	0=Non-Islam, 1=Islam
6. Wife's now working?	(binaire)	0=Yes, 1=No
7. Husband's occupation	(catégoriel)	1, 2, 3, 4
8. Standard-of-living index	(catégoriel)	1=low, 2, 3, 4=high
9. Media exposure	(binaire)	0=Good, 1=Not good
10. Contraceptive method used	(attribute de classe)	1=No-use 2=Long-term 3=Short-term

Dans la littérature (Lim, Loh et Shih, 2000), cette base est considérée comme étant l'une des plus difficiles pour la classification.

### La base de données 'Abalone'

La base de données 'Abalone' contient des mesures physiques de coquillages. L'ensemble de données contient des échantillons de 4177 instances avec 9 attributs (1 catégoriel et 8 numériques). Elle présente 29 classes.

Habituellement, l'âge du coquillage est déterminé par la découpe de la coquille au niveau du cône, après coloration, on compte le nombre d'anneaux à travers un microscope. Au lieu d'utiliser ce procédé invasif, des techniques de classification automatique sont utilisées pour prédire l'âge (longueur, diamètre, hauteur, poids total, poids écaillés, le poids des viscères, poids de la coquille).

Les données se présentent sur 10 descripteurs :

1)	Sex	catégoriel		M, F, et I (infant)
2)	Length	continue	en mm	Longest shell measurement
3)	Diameter	continue	en mm	perpendicular to length
4)	Height	continue	en mm	with meat in shell
5)	Whole weight	continue	en grammes	whole abalone
6)	Shucked weight	continue	en grammes	weight of meat
7)	Viscera weight	continue	en grammes	gut weight (after bleeding)
8)	Shell weight	continue	en grammes	after being dried
9)	Rings	entier		+1.5 gives the age in years
10)	Class	catégoriel		1 à 29

Pour cette base, les méthodes de classification classiques ont donné des résultats moyens.

### Les connaissances de similarité

Nous avons construit une heuristique de similarité pour les valeurs nominales (voir l'exemple dans les tableaux 5.1 et 5.2), afin de permettre le calcul de la fonction objective. L'heuristique est basée sur le principe de chevauchement (voir (Boriah, Chandola et Kumar, 2008)). La mesure de chevauchement consiste à compter le nombre d'attributs qui ont une correspondance dans les deux instances de données. La gamme de similarité par attribut pour la mesure du chevauchement est [0; 1]. La

valeur de 0 est donnée lorsque il n'ya pas de correspondance, la valeur de 1 est donnée lorsque les valeurs d'attribut sont identiques et toute autre valeur entre 0 et 1 est donnée lorsque la correspondance est de qualité moyenne.

**Tableau 5.1 Exemple d'heuristique de similarité pour l'attribut catégoriel 'sex' pour la base 'Abalone'.**

	Female	Male	Infant
Female	1.00	0.00	0.50
Male	0.00	1.00	0.50
Infant	0.50	0.50	1.00

Un coquillage de sexe 'Infant' peut devenir 'Male' ou 'Female' de manière égale à l'âge adulte. D'où la valeur de similarité fixée à 0.5

**Tableau 5.2 Exemple d'heuristique de similarité pour l'attribut catégoriels 'race' pour la base 'Adult'. Le symbole '?' est utilisé pour les données manquantes.**

	White	Asian-Pac-Islander	Amer-Indian-Eskimo	Other	Black	?
White	1,00	0,00	0,00	0,25	0,00	0,12
Asian-Pac-Islander	0,00	1,00	0,50	0,25	0,00	0,12
Amer-Indian-Eskimo	0,00	0,50	1,00	0,25	0,00	0,12
Other	0,25	0,25	0,25	1,00	0,25	0,12
Black	0,00	0,00	0,00	0,25	1,00	0,12
?	0,12	0,12	0,12	0,12	0,12	1,00

Dans notre étude, nous avons considéré la valeur inconnue "?" comme ayant une faible correspondance avec toute autre valeur. Ce qui a permis un traitement uniforme et efficace comme on le verra dans le chapitre résultat.

### 5.3.2 Pour l'étape 3 : « un modèle de mémoire basé PSO intégrant le critère d'adaptabilité »

Lors de l'étape 3, nous avons utilisé l'approche PSO décrite à travers l'algorithme PSO mixte amélioré de la section 5.2.3 du chapitre 3. Avec comme fonction objective celle intégrant la sémantique RGR, décrite dans l'équation 5 du chapitre 3. Le système se base sur le modèle de connaissances décrit dans les sections 3.2.1 à 3.2.2.

#### 5.3.2.1 Base de données 'GUIDIET'

Il s'agit d'une base de cas propriétaire utilisée dans une application permettant de proposer un menu diététique à une personne selon son profil : âge, sexe, dépense énergétique, IMC (indice de masse corporelle). Cette recommandation est faite en adaptant des cas constitués d'anciens profils accompagnés de leurs menus. La base des cas GUIDIET intègre 400 cas.

À partir d'un formulaire contenant un ensemble de descripteurs (Age, Sexe, Poids, Taille, Activité), ainsi qu'un ensemble de maladies (pour déterminer les antécédents médicaux de l'utilisateur) qui influencent les régimes alimentaires ou affectent le système gastrique, et en utilisant la base GUIDIET, un système basé cas peut déterminer l'état nutritionnel de l'utilisateur (en calculant l'IMC et la dépense énergétique et en considérant les antécédents médicaux). Il peut fournir un calendrier d'une semaine (de trois repas/jour) contenant des menus convenant à l'utilisateur.

Description du cas :

Le cas est décrit par les dimensions suivantes :

- *Âge* : l'âge de l'individu qui sera classé ensuite dans l'un des intervalles suivants : [1,3], [4,6], [7,12], [13,20], [21,45], [46,115].
- *Sexe* : le sexe de l'individu, masculin ou féminin

- *IMC* : l'indice de masse corporelle de l'individu : [10,20.5[, [20.5, 25],] 25,28], [28,40[
- *Dépense énergétique (DER)* : le nombre de calories dépensées par l'individu : [500,1200],] 1200,2500], ] 2500, 3000],] 3000,5000].
- *Antécédents médicaux* : prennent les valeurs 1 pour existants et 0 pour non existants.
- *Menu* : un calendrier d'une semaine qui comporte trois repas par jour: Petit\_déjeuner, Déjeuner, Dîner.

Chacun de ces repas est constitué à partir d'un ensemble d'aliments, que l'on catégorise comme suit :

- Les féculents.
- Les produits laitiers.
- Les boissons.
- Fruits.
- Légumes
- Matière grasse.
- Viande/Œuf
- Poissons
- Confiseries
- Divers : Pizza, Sandwich, ...

Chaque catégorie contient un ensemble d'aliments classés selon ses constituants en nutriments et leurs valeurs énergétiques. Cette structure facilite l'adaptation par la suite, et permet la construction de nouveaux menus et repas en choisissons les aliments préférés et conseillés pour l'état présenté.

Tableau 5.3 Exemples de cas dans GUIDIET

	Cas 1	Cas 2	Cas 3
Sexe	M	F	F
Age	2	30	58
IMC	22	27	24
Dépenses énergétiques	750	2800	1200
Diabète	0	1	0
Ulcère	0	0	0
Hypercholestérolémie	1	1	0
Goutte	0	0	0
Gastrite	0	0	0
Constipation	0	0	0
Intolérance au gluten	0	0	1
Ballonnements intestinaux	0	0	0
Grossesse	0	0	0
Colite spasmodique	0	0	0
Menu	1	3	4

## 5.3.2.2 Les connaissances utilisées

## Les connaissances de similarité

Concernant les descripteurs numériques, nous utiliserons la métrique normalisée de Grower (Diday et al., 1982) pour la distance entre deux descripteurs:

$$d_k(des_k^i, des_k^j) = 1 - \frac{|des_k^i - des_k^j|}{R_k} \quad (8)$$

où :

$R_k$  : est l'écart maximum entre les valeurs prises par le descripteur  $des_k$ . Toutes les distances  $d_k$  prennent des valeurs comprises entre 0 et 1 et on peut affecter à tous les  $\alpha_k$  la valeur 1. Ainsi, plus la valeur de  $D(cas_i, cas_j)$  est grande plus les  $cas_i$  et  $cas_j$  sont proches.

$des_k^i$  : valeur du  $k^{ième}$  descripteur pour le cas  $i$ .

$des_k^j$  : valeur du  $k^{ième}$  descripteur pour le cas  $j$ .

Pour les descripteurs symboliques, les heuristiques de similarité sont basées sur le principe suivant : la valeur de 0 est donnée lorsque il n'ya pas de correspondance, la valeur de 1 est donnée lorsque les valeurs d'attribut sont identiques:

#### Connaissances d'adaptation :

L'adaptation du cas en cours d'étude s'effectue sur le menu proposé, en appliquant des règles prédéfinies conçues pour fournir les équivalences des aliments en qualité ou en quantités.

Dans ce qui suit, nous donnons un exemple illustratif.

#### Heuristique d'adaptation pour l'attribut Diabète

Il faudrait penser à l'équilibre alimentaire qui est la base du traitement. Pour cela, il faut respecter les répartitions journalières des aliments pour maintenir une bonne glycémie, donc il faut modifier le menu de sorte qu'il préserve sa valeur énergétique.

*Exemple:*

Si (Maladie = "Diabète") Alors Substituer [(Prod\_lait. Concentré sucré) par (Prod\_lait. Ecrémé)].

Si (Maladie = "Diabète") Alors Substituer [(Viande. Rissolettes) par (Viande. Toutes les viandes)].

Si (Maladie = "Diabète") Alors Substituer [(Féculent. Chausson) par (Féculent. Pain blanc)].

Si (Maladie = "Diabète") Alors Substituer [(Féculent. Pain au chocolat) par (Féculent. Pain seigle)].

Si (Maladie = "Diabète") Alors Substituer [(Fruits. Fruits au sirop) par (Fruits. Fruits frais)].

Si (Maladie = "Diabète") Alors Substituer [(Boisson. Boissons fruités) par (Boisson. Eau Minérale)].

Le menu proposé n'est plus le cas extrait, mais il a la même structure et la même quantité énergétique fournie.

#### *Heuristique d'adaptation pour l'attribut IMC*

Il faut savoir qu'il n'existe aucun traitement ou régime standard, en raison de l'hétérogénéité clinique et biologique de l'affection liée à la surcharge pondérale : les attitudes thérapeutiques doivent donc être personnalisées, et un régime visant à une perte importante de poids ne doit être entrepris que sous suivi médical et psychologique.

#### *Exemple:*

Si (IMC>29) Alors Substituer [(Viande. Entrecôtes) par (Viande. Bœuf)].

Si (IMC>29) Alors Substituer [(Viande. côtelettes) par (Viande. Jambon dégraissé)].

Si (IMC>29) Alors Substituer [(Féculent. Pain au chocolat) par (Féculent. Pain blanc)].

Si (IMC>29) Alors Substituer [(Féculent. Brioche) par (Féculent. Pain complet)].

### Les connaissances d'adaptabilité :

Après avoir dégagé les connaissances d'adaptation, on doit déterminer les coûts d'adaptabilité qui sont des scores établis sur la base du nombre d'opérateurs d'adaptation applicables qui indique la dépendance entre le problème et la solution. Le tableau 5.4 présente quelques exemples illustratifs:

**Tableau 5.4 Dépendances Partielles Problème/solution dans GUIDIET**

Sol.	Pbm.					
	IMC > 29	Diabète	Ulcère	Hypercholestérolé mie	Gluten intolerance	Grossesse
Produits Laitiers	√			√	√	√
Sodas		√			√	
Féculents	√	√		√	√	
Confiserie		√	√	√	√	√
Poissons	√			√		
Viandes/Œufs	√	√		√		
Fruits				√		
Légumes				√		
Matières grasses	√			√		√
Divers (pizza, sandwich...)	√	√		√	√	√

Pour déterminer le coût d'adaptabilité, il faut d'abord quantifier les dépendances: par exemple, dans quelle mesure la contrainte 'diabète' influence le choix des aliments, cette information se déduit aisément de la table de dépendance préétablie (Tableau 5.4). L'exemple suivant illustre deux situations avec des coûts d'adaptabilité différents:

#### *Situation 1*

Le problème source présente des dissemblances avec le problème cible sur les descripteurs 'IMC>29' et 'Grossesse'. Ainsi, les coûts d'adaptabilité partiels sont les

suivants:

$\mu$  (IMC>29) = 6 (nombre de lignes cochées lors dans la colonne 'IMC>29' du tableau VII.3)

$\mu$  (Grossesse) = 4 (nombre de lignes cochées lors dans la colonne 'Grossesse')

Le coût total de l'adaptation est:  $\mu$  (cas source) = 10.

### *Situation 2*

Le problème source présente des dissemblances avec le problème cible sur les descripteurs de 'Diabète' et 'Hypercholestérolémie'. Les coûts d'adaptabilité partiels sont les suivants:

$\mu$  (Diabète) = 5

$\mu$  (Hypercholestérolémie) = 9

Le coût total de l'adaptation est:  $\mu$  (cas source) = 14

Même si dans les deux cas la différence entre le problème source et le problème cible est sur deux descripteurs, le cas source dans la situation 1 est plus facile à adapter que celui de la situation 2.

### 5.3.3 Étape 4 : « une modélisation CBR pour la planification »

Lors de l'étape 4, nous avons utilisé l'approche PSO décrite à travers l'algorithme PSO mixte amélioré de la section 5.2.3 du chapitre III. Avec comme fonction objective celle intégrant la sémantique RGR, décrite dans l'équation 5 du chapitre 3. Le tout basé sur le modèle de connaissances décrit dans le chapitre 4 et partiellement reprises dans les sections 3.3.1. à 3.3.2.

## 5.3.3.1 Base de 'PLAN'.

Il s'agit d'une base de cas non publique utilisée à titre illustratif dans une application permettant de proposer un itinéraire alternatif. La base des cas PLAN intègre 94 cas, représentant chacun un itinéraire dans la zone urbaine du centre-ville de Montréal délimitée par les quatre intersections suivantes :

- Coin Sherbrooke / de Bleury
- Coin Sherbrooke / Berri
- Coin Berri / Notre Dame
- Coin Notre Dame / de Bleury

Nous avons pu identifier 116 intersections importantes (identifiables par la numérotation du tableau 5.5) délimitant les segments de chemin.

**Tableau 5.5 Numérotation des intersections pour la base PLAN**

Intersection . Numéro.	Longitude	Latitude * -1	Rue 1	Rue 2
1	45.508409	73.571364	Sherbrooke	de Bleury
2	45.509348	73.570805	Sherbrooke	Jeanne Mance
3	45.510401	73.570355	Sherbrooke	Kimberley
4	45.511081	73.570054	Sherbrooke	Saint-Urbain
5	45.511981	73.569648	Sherbrooke	Clark
6	45.512537	73.569412	Sherbrooke	Saint-Laurent
...	...	...	...	...
38	45.507552	73.569573	de Bleury	Kennedy
39	45.508379	73.568779	Kennedy	Jeanne Mance
40	45.509334	73.56791	Kennedy	Kimberley
...	...	...	...	...

Description du cas :

Le cas est décrit par les dimensions suivantes :

Tableau 5.6 Exemples de cas dans PLAN

id_cas	Pref	Ori	Dest	dist/km	durée/m	n	nb_inters	Int <sub>1</sub>	...	int <sub>i</sub>	...	int <sub>n</sub>
1	0	1	107	2,2	6	18	1	...	5	...	107	
2	2	113	13	2	6	19	113	...	56	...	13	
3	2	114	39	2	6	20	114	...	101	...	39	
4	3	39	114	1,9	5	17	39	...	107	...	114	
5	1	107	1	1,8	5	17	107	...	92	...	1	

On notera que la description du cas introduite à ce niveau reprend l'essentiel de celle du niveau modélisation pour ce qui est du contenu, mais non pour ce qui est de la hiérarchisation. Le fait est qu'une représentation plate du cas est plus pragmatique.

### 5.3.3.2 Les connaissances utilisées

#### Les connaissances de similarité

Comparer des cas revient à comparer des intersections (symbolique ou numérique), des préférences (symbolique), des distances et des durées (numériques).

Les intersections sont représentées ici de manière symbolique par un nombre inclus entre [1,116]. À chaque symbole sont associées la latitude et la longitude correspondant à l'intersection. On pourrait donc penser utiliser une métrique de distance afin de mesurer la similarité entre deux intersections. Or cette façon de procéder ne reflète pas la connectivité des intersections. Nous avons donc opté pour une heuristique dans le style mesure de chevauchement (voir tableau 5.7).

**Tableau 5.7 Exemple d'heuristique de similarité pour les dimensions de type 'intersection'.**

	1	2	3	38	40
1	1,00	0,95	0,90	0,95	0,00
2	0,95	1,00	0,95	0,00	0,00
3	0,90	0,95	1,00	0,00	0,95
38	0,95	0,00	0,00	1,00	0,90
40	0,00	0,00	0,95	0,90	1,00

Rappelons que la gamme de similarité par attribut pour la mesure du chevauchement est  $[0; 1]$ . La valeur de 0 est donnée lorsque il n'ya pas de correspondance, la valeur de 1 est donnée lorsque les valeurs d'attribut sont identiques et toute autre valeur entre 0 et 1 est donnée lorsque la correspondance est de qualité moyenne.

L'heuristique de similarité pour les préférences est binaire : elle aura la valeur 1 lorsque les valeurs sont identiques et 0 sinon.

Quant aux descripteurs numériques, leur comparaison se fera selon une métrique de distance classique.

### ***Les connaissances d'adaptabilité***

L'évaluation de l'adaptabilité d'un plan se fait sur sa partie solution qui est en l'occurrence le chemin constitué de toutes les intersections menant de la source à la destination.

Pour déterminer le coût d'adaptabilité, il faut d'abord quantifier l'importance de chaque intersection du chemin à adapter: par exemple, dans quelle mesure modifier l'intersection '103' (correspondant à l'intersection Viger / entrée 5 de l'autoroute 720) influence le coût du chemin généré. Cette importance peut être représentée par

la fréquentation de cette intersection ou en d'autres termes la fréquence d'occurrence de cette intersection dans tous les chemins de la base de cas. Ainsi :

$$\mu(int_i) = \text{Fréquence de } int_i \text{ dans Base-Plan.}$$

Les valeurs de la fonction  $\mu$  sont préétablies pour l'ensemble des intersections dans une phase de prétraitement et seront utilisées lors de la phase de rappel.

#### 5.4 Environnement de programmation

Précisons que l'objectif de la programmation est le développement de nucléus d'application à des fins de test et validation des choix méthodologiques et non d'outils livrables à l'utilisateur final.

Notons par ailleurs que le contexte d'intégration de nos outils étant les systèmes de transport intelligent (STI), ces derniers imposent leurs propres normes de programmation et d'intégration. À ce titre, il aurait été inutile de chercher à intégrer l'ensemble des nucléus développés dans un même environnement offrant une interface utilisateur (cette dernière ayant elle même sa problématique de modalité : dispositif général ou spécialisé).

Nous avons donc opté pour un environnement de programmation qui priorise :

- l'efficacité de calcul notamment le calcul matriciel,
- la souplesse de programmation
- la facilité d'intégration de code provenant d'autre langage de haut niveau tel que C++ ou Java.
- La présence d'un grand nombre de toolbox facilitant différents calculs.
- La présence d'outils intégrés de visualisation des résultats.

L'environnement Matlab dans sa version 7.6 nous a donc pleinement satisfaits pour les exigences de cette partie.

Par ailleurs, l'environnement physique est doté d'un processeur de type AMD Athlon™ 64x2 processeur Dual-Core 3800+, cadencé à une fréquence d'horloge de 2,00 GHz, avec 2,90 Go de RAM.

### 5.5 Conclusion

Dans une certaine mesure, la diversité des expérimentations planifiées dans le présent protocole, qu'elles soient directes ou transversales nous assure la robustesse de ce dernier, même si les applications choisies sont d'ordre académique et non industriel.

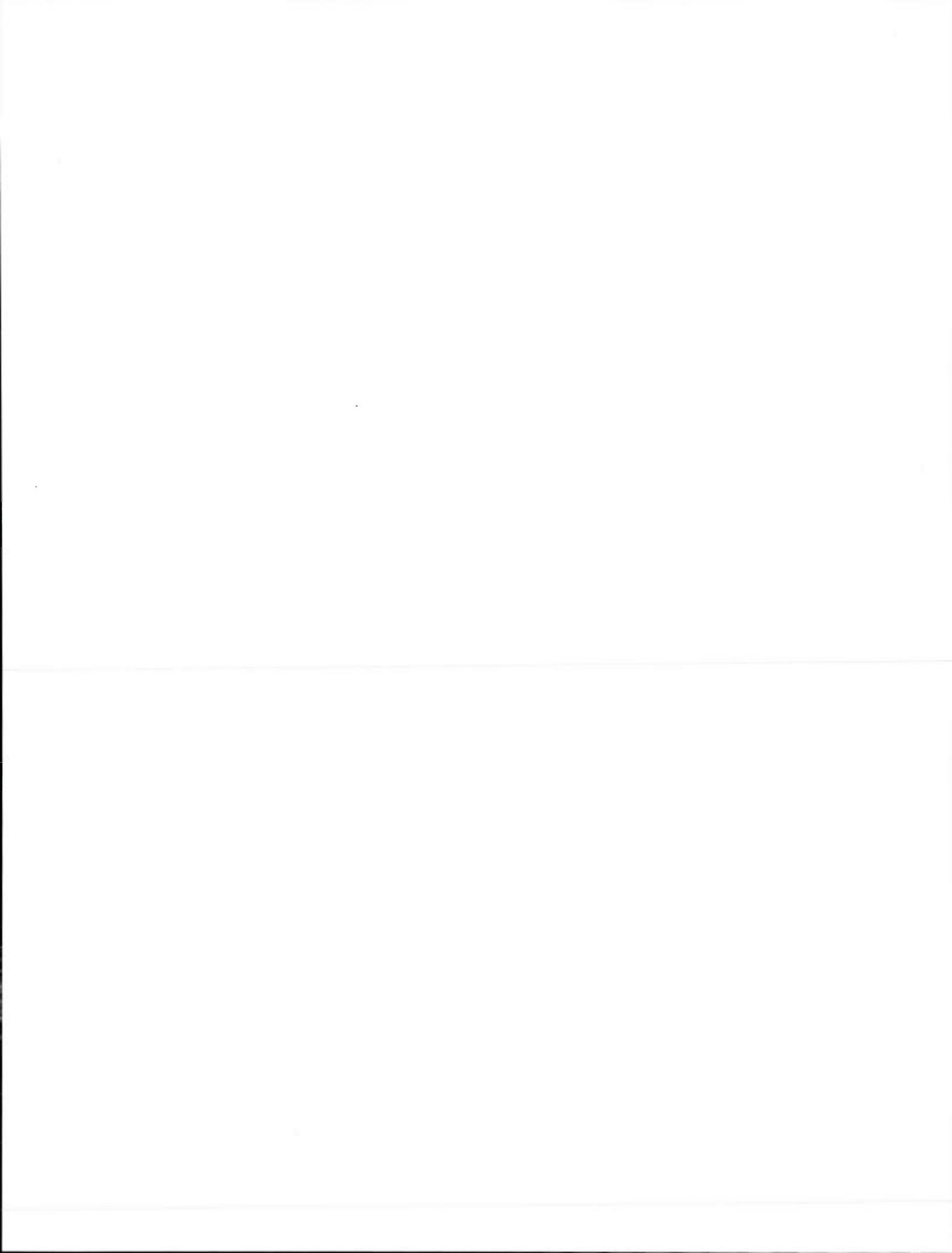
C'est d'ailleurs le manque de données de taille réelle qui nous a conduits vers une telle démarche dont la richesse ne fait que confirmer l'extensibilité de nos approches à d'autres horizons.

Nous allons voir dans le chapitre VI, comment les bases de données présentées ainsi que les connaissances de similarité et d'adaptabilité (le cas échéant) sont utilisées par les différentes approches PSO soit lors de la classification, soit de la recherche par similarité soit de la recherche guidée réutilisation.

## CHAPITRE VI

# Résultats et discussion

1. Introduction.
2. Validation des mécanismes d'interprétation.
3. La PSO pour la recherche en mémoire.
4. Discussion.
5. Conclusion.



## 6.1 Introduction

Les expérimentations menées ont donné lieu à des résultats intéressants, voir même très encourageants pour certains aspects. Nous présentons dans la suite, les résultats obtenus pour chacune des étapes d'expérimentation.

## 6.2 Validation des mécanismes d'interprétation

Par manque de données dans le domaine du CBP pour une étude comparative, nous avons procédé à une validation transversale en faisant appel à une tâche de classification<sup>15</sup>. Nous avons utilisé trois bases de données extraites de UCI Repository (Asuncion et Newman, 2007).

Pour la base 'Adult', nous avons effectué une validation croisée en 10 rotations sur les 48842 instances. Les données ont été réparties en 90% des instances pour l'apprentissage et 10% des instances pour le test, de telle sorte que la fréquence des classes reste homogène dans l'ensemble des 10 rotations. L'algorithme PSO a été appliqué à dix reprises et nous avons considéré la moyenne des exécutions.

La précision de reconnaissance moyenne était de  $89,84 \pm 0,005\%$  pour une population de 10 particules, avec un temps d'apprentissage de 4,45 heures.

Un résultat substantiellement meilleur a été obtenu lorsque la taille de la population de particules a été augmentée à 40 particules. La précision de reconnaissance moyenne est passée à  $99,56\% \pm 0,005$ . Toutefois, le temps de calcul a bondi à 17,74 heures, montrant que dans la pratique, un compromis doit être fait entre la précision et le temps de d'apprentissage, selon le contexte.

Pour la base CMC, nous avons donc effectué une validation croisée en 10 rotations sur les 1473 instances. Les données ont été réparties en 90% des instances pour

---

<sup>15</sup> Les détails de la classification basée PSO sont présentés en appendice A.

l'apprentissage et 10% des instances pour le test, de telle sorte que la fréquence des classes reste homogène dans l'ensemble des 10 rotations. L'algorithme PSO a été appliqué à dix reprises et nous avons considéré la moyenne des exécutions.

La moyenne de la précision est de  $94.58\% \pm 0.027$  pour une population de 10 particules, avec un temps d'apprentissage de 22.30 minutes. Lorsque l'on fait varier la taille de l'essaim (voir résultats détaillés en Appendice A), la précision continue à augmenter ainsi que le temps d'apprentissage.

Quant à la base 'Abalone', nous avons effectué une validation croisée en 10 rotations sur les 4177 instances. Les données ont été réparties en 90% des instances pour l'apprentissage et 10% des instances pour le test, de telle sorte que la fréquence des classes reste homogène dans l'ensemble des 10 rotations. L'algorithme PSO a été appliqué à dix reprises et nous avons considéré la moyenne des exécutions. La moyenne de la précision est  $94.41\% \pm 0.027$  pour une population de 10 particules, avec un temps d'apprentissage de 1.24 heure.

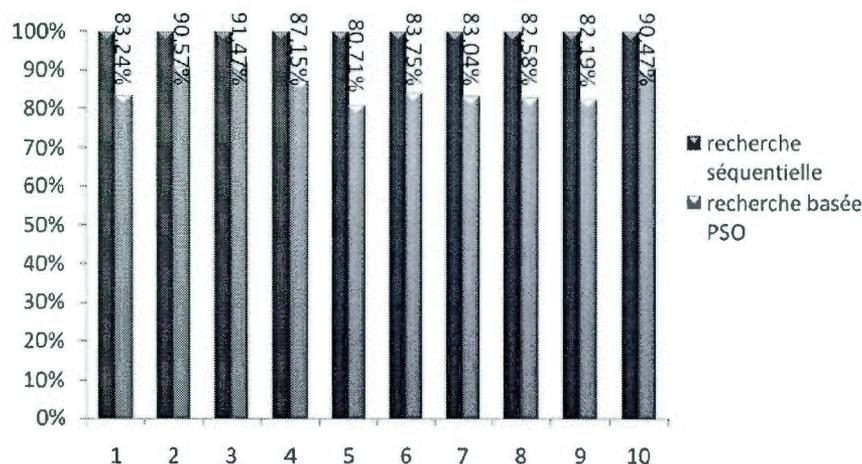
### 6.3 Résultats de la PSO pour la recherche en mémoire

#### 6.3.1 Résultats de la recherche basée similarité

Dans cette série d'expérimentations, nous avons utilisé la base de données 'Adult' comme base de cas. Chacune de ses 48842 instances a été considérée comme un cas.

Tous les résultats présentés dans cette section représentent la moyenne de 10 essais. Les paramètres PSO ont été fixés à  $v_{max}=0.5$ ,  $v_{min}=-0.5$ ,  $c_1=1.7$ ,  $c_2=1.7$ ,  $w_{max}=0.9$ ,  $w_{min}=0.4$ , le nombre de particules = 10 et  $T_{max} = 1000$ . Les Figures 6.1 à 6.3 reprennent les résultats obtenus.

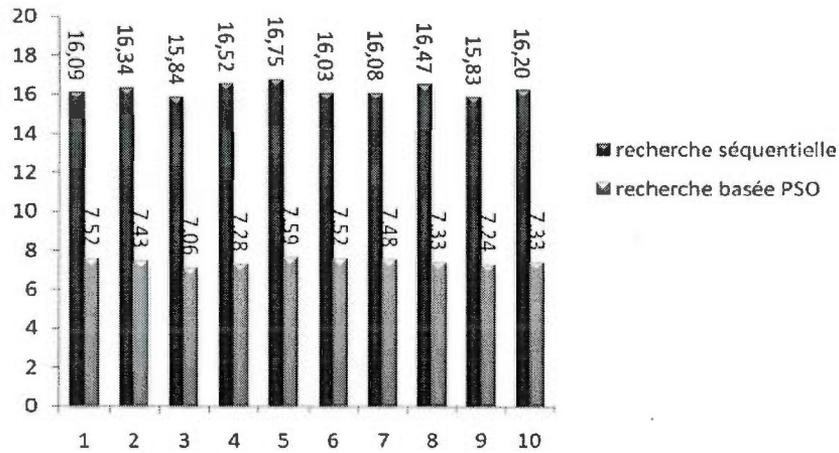
À des fins de comparaison, nous avons effectué une recherche séquentielle<sup>16</sup> avec les mêmes requêtes. Nous avons d'abord choisi dix requêtes pour lesquelles il existe une correspondance parfaite (similarité égale à 1) dans la base de cas. Comme prévu, la caractéristique d'exhaustivité de la recherche séquentielle a conduit à une extraction parfaite de ces cas (Figure 6.1). La précision de rappel pour l'algorithme PSO a varié entre 80,71% et 91,47%, en fonction de la requête.



**Figure 6.1** Étude comparative de la similarité (axe des y) entre recherche linéaire et recherche basée PSO pour 1<sup>ie</sup> groupe de requêtes (axe des x).

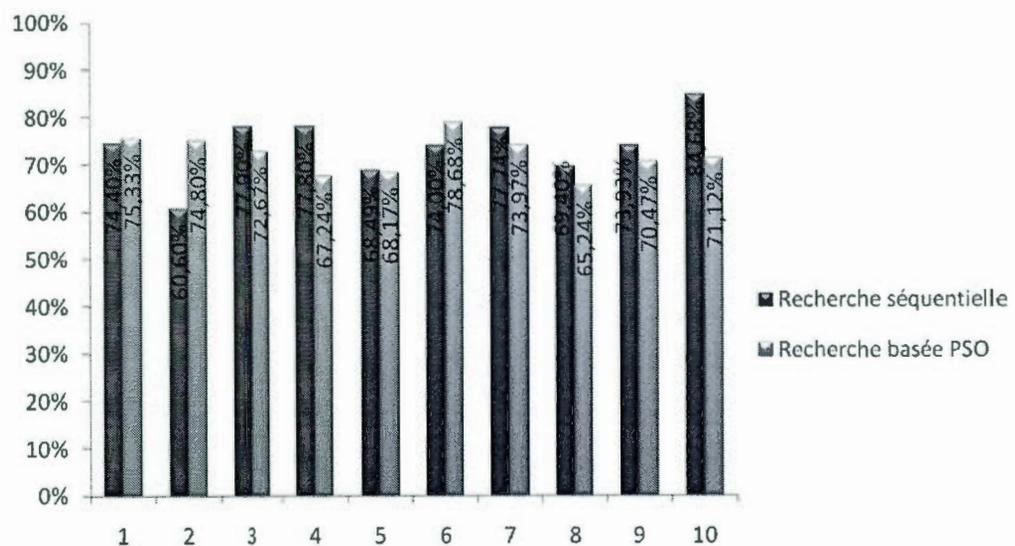
Nous avons aussi comparé le temps de calcul des deux approches. La figure 6.2 montre que la recherche basée PSO réduit le temps de recherche de près de la moitié en comparaison à la recherche linéaire.

<sup>16</sup> L'alignement sur la recherche séquentielle n'est pas un choix fortuit. Il s'agit de l'approche de recherche exhaustive sur laquelle toute étude comparative s'aligne notamment sur le plan précision.



**Figure 6.2** Étude comparative du temps de recherche (en secondes sur l'axe des y) de la recherche linéaire et de la recherche basée PSO pour le 1<sup>er</sup> groupe de requêtes (axe des x).

Nous avons ensuite choisi dix requêtes pour lesquelles il existe une similarité moins parfaite dans la base de cas (pour éviter la situation de remémoration parfaite). La caractéristique d'exhaustivité de la recherche séquentielle devrait également être en mesure de récupérer le cas le plus similaire (Figure 6.3), mais nous remarquons que la recherche PSO donne de meilleurs résultats pour certaines requêtes (1, 2 et 6). Cela signifie que le voisinage artificiel généré par la PSO a une qualité de couverture meilleure que le voisinage disponible dans la base. En d'autres termes, la couverture de la base de cas n'est pas assez riche. Le voisinage généré par la PSO est un bon indicateur de la qualité de la couverture de la base de cas. Il pourrait par ailleurs servir pour l'enrichissement de la base de cas.



**Figure 6.3** Étude comparative de la similarité (axe des y) de la recherche linéaire et de la recherche PSO pour le 2<sup>nd</sup> groupe de requêtes (axe des x).

Le temps pour la recherche PSO reste entre 6 secondes et 8 secondes, et est toujours approximativement réduit de moitié pour la recherche linéaire.

### 6.3.2 Résultats de la PSO et la RGR

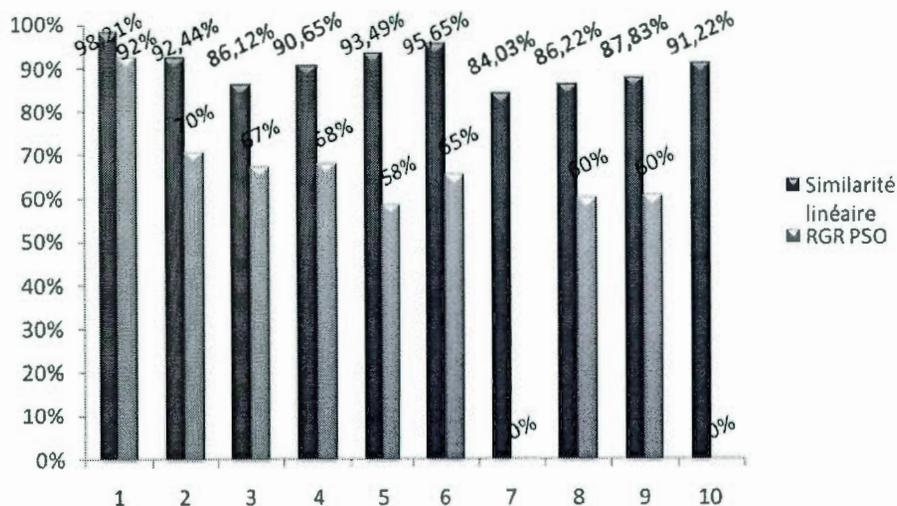
Dans cette série d'expérimentations, nous avons utilisé la base de données 'GUIDIET' comme base de cas. Les requêtes présentées au systèmes sont décrites dans le tableau 6.1.

Tableau 6.1 Requêtes utilisée pour la base GUIDIET.

Requête	Age	Sexe	IMC	Dépense Énergétique	Antécédents Médicaux
#1	48	M	20,45	2540	Null
#2	18	F	19	1986	Ulcère, grossesse
#3	5	M	20	500	Diabète, Ulcère, Intolérance au Glutène
#4	60	M	22	1200	Diabète, Ulcère, Gastrite
#5	33	M	16	1800	Ulcère, Gastrite, Colite Spasmodique
#6	28	F	24	2400	Diabète, Grossesse
#7	23	F	18	1600	Diabète, Hyper-Cholesteremie, Gastrite, Ballonnement, Constipation, Grossesse
#8	85	F	22	1850	Ulcère, Ballonnement, Goute
#9	15	F	19	600	Diabète, gastrite, Intolérance au Gluten
#10	50	F	24	3000	Gastrite, Collite Spasmodique, Goute, Grossesse

Les paramètres PSO ont été fixé à  $v_{max}=0.5$ ,  $v_{min}=-0.5$ ,  $c_1=2.0$ ,  $c_2=2.0$ ,  $w_{max}=0.9$ ,  $w_{min}=0.4$ , le nombre de particules = 5 et  $T_{max} = 100$ . La Figures VI.4 reprend les résultats obtenus.

Nous avons choisi dix requêtes pour lesquelles il existe une similarité non parfaite dans la base de cas (pour éviter la situation de remémoration parfaite). La caractéristique d'exhaustivité de la recherche séquentielle a été en mesure de récupérer le cas le plus similaire (Figure 6.4).



**Figure 6.4** Similarité par recherche linéaire et recherche guidée réutilisabilité basée PSO pour des requêtes non existantes dans la base de cas (axe des x).

Pour ces mêmes requêtes nous avons procédé à une recherche guidée par la réutilisabilité greffée sur un algorithme PSO. Sur la courbe RGR PSO de la figure 6.4, une valeur  $\geq 90\%$  sera interprétée comme une valeur de similarité alors que les valeurs  $< 90\%$  sont interprétées comme un coût de réutilisabilité du cas trouvé.

Nous noterons que pour les requêtes 7 et 10, aucun voisinage PSO n'a pu être généré. Un autre résultat pertinent est le fait que les cas 76 et 167 apparaissent comme des cas candidats à la réutilisation pour différentes requêtes avec un coût variable. Nous commenterons ces résultats en détail dans la section discussion.

### 6.3.3 Résultats de la PSO pour le CBP

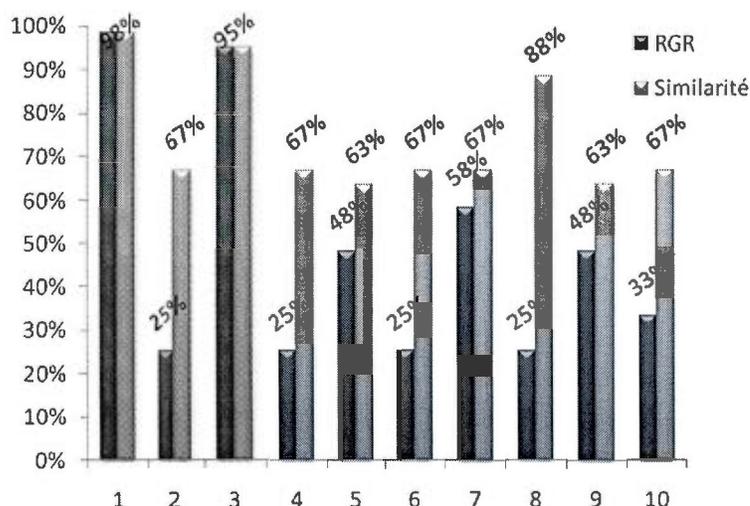
Dans cette série d'expérimentations, nous avons utilisé la base de données 'PLAN' comme base de cas. Les requêtes présentées au système sont décrites en tableau 6.2.

**Tableau 6.2 Requêtes pour la base PLAN.**

Requête	Origine	destination	Préférence
#1	1	108	Nulle
#2	39	55	Chemin usuel
#3	104	1	Chemin usuel
#4	87	13	Chemin usuel
#5	101	93	Chemin sans virage
#6	112	46	Chemin usuel
#7	41	107	Chemin sans virage
#8	76	32	Chemin usuel
#9	101	93	Chemin sans virage
#10	78	39	Chemin usuel

Les paramètres PSO ont été fixé à  $v_{max}=0.5$ ,  $v_{min}=-0.5$ ,  $c_1=2.0$ ,  $c_2=2.0$ ,  $w_{max}=0.9$ ,  $w_{min}=0.4$ , le nombre de particules = 5 et  $T_{max} = 100$ . La Figure 6.5 résume les résultats obtenus.

Nous avons choisi dix requêtes pour lesquelles il existe une similarité non parfaite dans la base de cas (pour éviter la situation de remémoration parfaite). La caractéristique d'exhaustivité de la recherche séquentielle a été en mesure de récupérer le cas le plus similaire (Figure 6.5).



**Figure 6.5 Recherche linéaire guidée similarité et recherche guidée réutilisabilité basée PSO pour des requêtes non existantes dans la base de cas (axe des x).**

Pour ces requêtes nous avons procédé à une recherche guidée par la réutilisabilité greffée sur un algorithme PSO. Sur la courbe RGR PSO de la figure 6.5, une valeur  $\geq 90\%$  sera interprétée comme une valeur de similarité alors que les valeurs  $< 90\%$  sont interprétées comme un coût de réutilisabilité du cas trouvé.

#### 6.4 Discussion

L'aspect inédit de l'approche de recherche basée PSO fait qu'aucun travail de référence n'a pu être trouvé dans la littérature. Nous avons donc opté pour une comparaison avec la méthode de référence qu'est la recherche linéaire ou séquentielle.

Nous avons d'abord choisi dix requêtes pour lesquelles il existe une correspondance parfaite (similarité égale à 1) dans la base de cas. Comme prévu, la caractéristique d'exhaustivité de la recherche linéaire a conduit à une extraction

parfaite de ces cas. La précision de rappel de la PSO, quant à elle, a varié entre 80,71% et 91,47%, en fonction de la requête pour une consommation du temps de recherche de moitié moindre par rapport à la recherche linéaire.

Nous avons ensuite choisi dix requêtes pour lesquelles il existe une similarité moins parfaite dans la base de cas (pour éviter la situation de remémoration parfaite). La caractéristique d'exhaustivité de la recherche linéaire a permis la récupération des meilleurs cas similaires. Dans ce cas de figure, la recherche PSO a donné de meilleurs résultats pour certaines requêtes. Cela se traduit par le fait que le voisinage artificiel est plus proche des requêtes que le voisinage réel. La base de cas a donc une qualité de couverture moindre.

Notons que le voisinage généré par la PSO est un bon indicateur de la qualité de la couverture de la base de cas. Il pourrait être utilisé pour l'enrichissement de la base de cas.

Le temps de remémoration a été là aussi environ de moitié inférieur à celui de la recherche linéaire.

### *Analyse de la complexité algorithmique*

Nous procédons à l'analyse de la complexité algorithmique de l'approche PSO appliquée à la recherche des données mixtes en la situant par rapport à d'autres approches de recherche en mémoire. Soient:

- $M$  : le nombre de particules de l'essaim,
- $N$  : le nombre de dimensions de l'espace problème,
- $T_{max}$  : le nombre maximum d'itérations,

Définissons:

- $Coût_1$  : le coût de l'initialisation des positions et leurs vitesses,
- $Coût_2$  : le coût de calcul de la fonction fitness (défini par  $N * Coût_5$ ),
- $Coût_3$  : le coût de mise à jour de la meilleure particule globale,

- $Coût_4$  : le coût de mise à jour de la particule courante,
- $Coût_5$  : le coût élémentaire de la fonction fitness,
- $Coût_6$  : le coût d'interprétation.

Nous pouvons alors calculer la complexité pour l'algorithme A3 (donné en figure III.8) la recherche des données mixtes, par:

$$\begin{aligned}
 Coût_{A3} &= M * Coût_1 + T_{max} * M * Coût_6 + T_{max} * M * Coût_2 + T_{max} * M * Coût_3 + T_{max} * Coût_4 \\
 &= M * Coût_1 + T_{max} * M * Coût_6 + T_{max} * M * N * Coût_5 + T_{max} * M * Coût_3 + T_{max} * \\
 &Coût_4
 \end{aligned}$$

On peut constater à partir de cette équation que  $Coût_{A3}$  a une complexité de l'ordre de  $O(T_{max} * M * N)$ . Par ailleurs, on note que les facteurs desquels dépend le coût comprennent  $T_{max}$  et  $M$  qui sont inhérents à l'algorithme, et  $N$  qui est inhérent au problème.

En dépit du fait que  $M \ll T_{max}$ , nous ne pouvons négliger  $M$  puisqu'il s'agit de facteurs multiplicateurs de  $T_{max}$ . En conséquence, un compromis doit être envisagé lors du choix des valeurs de  $T_{max}$  et  $M$ , puisque  $N$  est hors de notre contrôle.

Le fait d'avoir le contrôle sur deux paramètres parmi 3, rend cette approche de recherche en mémoire plus qu'intéressante comparativement à la recherche linéaire qui est de l'ordre  $O(Taille)$ , Taille étant le nombre de cas dans la base de cas. C'est un paramètre hors de contrôle.

En comparaison, une recherche Fish & Shrink pour laquelle la complexité computationnelle dépend de la connectivité des aspects et de la taille de la requête, aucun des paramètres ne peut être contrôlé. De même pour un modèle aussi performant que les CRN, dont la complexité dépend de la taille de la requête, de la

connectivité des entités d'information et de leur spécificité, aucun de ces trois paramètres n'est sous contrôle.

En revanche, pour une approche telle que les k-d Trees, la complexité moyenne étant de l'ordre de  $O(\log_2 \text{Taille})$  selon (Wess, Althoff, et Derwand 1994), on serait tenté par la placer en tête des modèles de mémoires de cas. Or son incapacité à prendre en charge l'aspect apprentissage incrémentale du CBR la rend d'utilité peu probable.

Quant à la recherche PSO basée RGR, là aussi l'aspect inédit de cette approche fait qu'aucun travail de référence n'a pu être trouvé dans la littérature. Nous avons donc opté pour une comparaison avec la méthode de référence qu'est la recherche linéaire.

Nous avons donc choisi dix requêtes pour lesquelles il existe une similarité moins parfaite dans la base de cas. La caractéristique d'exhaustivité de la recherche linéaire a permis la récupération des meilleurs cas similaires.

Nous avons observé que pour les requêtes 7 et 10, aucun voisinage PSO n'a pu être généré. Lors d'une analyse plus approfondie, il s'avère que ces requêtes représentaient des situations relativement rares par rapport à la base de cas. En effet, il s'agissait de personnes cumulant un nombre important de pathologies.

Requête 7 : femme de 23 ans, diabétique présentant de l'hyper-ch, une gastrite, des ballonnements, de la constipation et une grossesse.

Requête 10 : femme de 50 ans ayant une consommation énergétique quotidienne de 3000 calories présentant une gastrite des colites spasmodiques, la goutte et une grossesse.

Ces situations étant si peu fréquentes, seule une exploration exhaustive de l'espace de recherche pourrait les détecter. Il s'agit de cas qualifiés d'«essentiels»<sup>17</sup> selon la taxonomie proposée dans (Fuch, 1997).

Un autre résultat pertinent est le fait que les cas 76 et 167 apparaissent comme des cas candidats à la réutilisation pour différentes requêtes avec un coût variable. Cela vient du fait que ces cas sont du type cas 'pont'.

Notons que l'analyse de la complexité algorithmique pour le cas de l'ajout de la RGR donne le même résultat puisque seul le *Coût<sub>5</sub>* : coût élémentaire de la fonction objective, va varier, or il n'apparaît pas dans l'expression finale de la complexité qui reste de l'ordre de  $O(T_{max} * M * N)$ .

Pour ce qui est de l'utilisation du modèle de mémoire basé PSO dans le cadre de la planification, là non plus aucun travail de référence n'a pu être trouvé dans la littérature. Nous avons, là aussi, opté pour une comparaison avec la méthode de référence qu'est la recherche linéaire.

Nous avons donc choisi dix requêtes pour lesquelles il existe une similarité moins parfaite dans la base de cas. La caractéristique d'exhaustivité de la recherche linéaire a permis la récupération des meilleurs cas similaires.

L'approche PSO a été en mesure de récupérer les cas les plus similaires lorsque la similarité était maximale, mais sinon indiquer le coût d'adaptation lorsque la similarité était en de ça des 90%. L'analyse de complexité de l'algorithme reste la même que précédemment.

---

<sup>17</sup> Se référer à la section 4.1. du chapitre I.

## 6.5 Conclusion

Les résultats diversifiés présentés dans cette partie nous conduisent aux constats suivants :

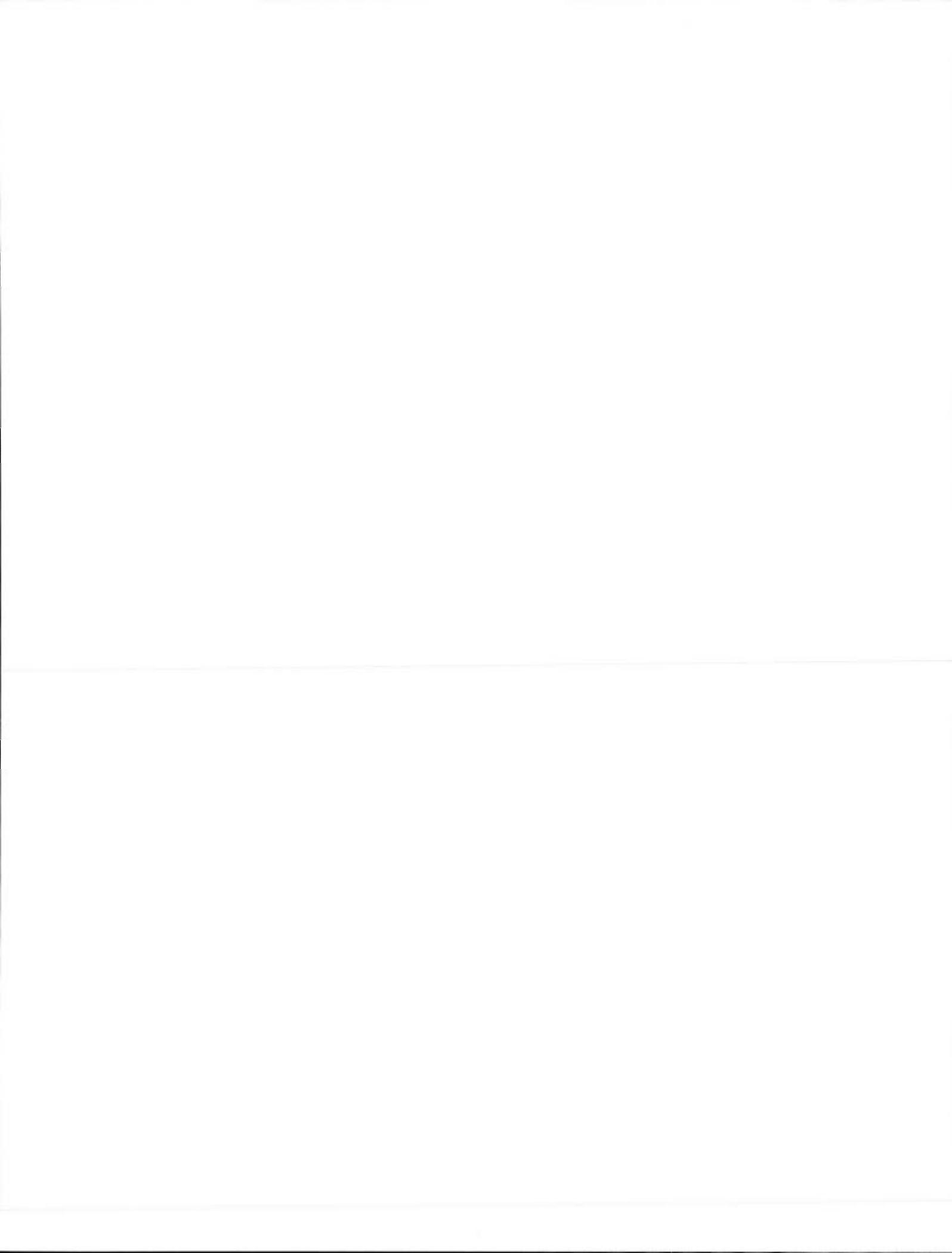
- La souplesse de la PSO à représenter des problèmes abordant des tâches de résolution différentes : classification, recherche en mémoire, planification.
- L'efficacité de cette métaheuristique en classification.
- Même lorsque le résultat en recherche n'est pas optimal comparativement à la recherche linéaire, la raison n'est pas inhérente à la méthode, mais à la qualité de couverture de la base.

À la lumière de ces constats, nous pouvons conclure que la PSO a des potentiels d'application multiples. C'est un terrain de recherche très fertile pour lequel nous n'avons fait qu'entrouvrir la porte.

## CHAPITRE VII

# CONCLUSION GÉNÉRALE

1. Rappel des objectifs
2. Évaluation et critiques
3. Contributions
4. Perspectives



## 7.1 Rappel des objectifs

Les objectifs fixés lors de ce travail étaient la mise en œuvre de solutions correspondant à différents énoncés que nous consolidons ici en un énoncé de thèse unique :

*La clé du succès de la résolution de problème de second principe est la remémoration du bon cas. Par bon cas, on entend celui qui rendra la réutilisation efficace, donc le plus similaire dans sa partie problème et le plus adaptable dans sa partie solution. L'élimination du biais structurel améliore la perception de l'espace de recherche et la qualité du rappel par là même. Le parcours stochastique de l'espace de recherche permet la génération du voisinage caractérisé par une réutilisabilité forte puis un accès direct à ce voisinage dans une mémoire associative de cas.*

Les objectifs de la thèse étaient décrits à travers une démarche de résolution incrémentale, à savoir :

Objectif 1 : Le développement d'un algorithme PSO pour la prise en charge de données mixtes. L'algorithme a la capacité de traiter une description de problème comportant des dimensions de nature mixte (continues, discrètes et catégorielles). La disponibilité des données et la diversité des travaux dans le domaine de la classification automatique nous ont permis de tester et valider l'approche avec grand succès. Le mécanisme ainsi validé est repris comme noyau au niveau de l'objectif 2.

Objectif 2 : La mise en œuvre d'un premier modèle de mémoire basé PSO. Il s'agissait d'un modèle de mémoire de cas qui respecte un critère de recherche basé similarité et qui est fondé sur une approche par essaim de particules pour la recherche en mémoire.

Objectif 3 : L'intégration du critère de réutilisabilité dans le modèle de mémoire basé PSO. On augmente les aptitudes de recherche de l'algorithme précédent en enrichissant la sémantique de la fonction objective au moyen de l'équation de transfert. On obtient ainsi une remémoration guidée par la réutilisation (RGR)

Objectif 4 : La représentation des connaissances CBR pour la planification. Il s'agit de mettre en place des modèles de connaissance nécessaires à la capture de l'expertise de génération de plan personnalisée, incluant le modèle de connaissances de similarité, d'adaptation et d'apprentissage.

Dans ce qui suit, nous faisons le point sur les résultats obtenus par rapport aux objectifs fixés.

## 7.2 Évaluation et critiques

De nombreux modèles de mémoire de cas ont été proposés dans la littérature. En comparaison, l'approche basée PSO nous paraît intéressante pour de multiples raisons :

- Convient à l'exploration d'un espace de recherche. C'est un comportement natif de la PSO. Nous l'avons juste adaptée au traitement des données mixtes.
- Utilise une représentation flexible des connaissances comparativement à d'autres approches stochastiques telles que les algorithmes génétiques, il n'y a pas de codage contraignant.
- Offre une bonne performance de calcul relativement à de multiples approches classiques.
- A une grande échelle d'applicabilité. A partir du moment où le critère de recherche peut s'exprimer en une fonction à optimiser, la PSO peut s'appliquer.

- Pertinence pour l'apprentissage de nouveaux cas. En fait, l'ajout d'un nouveau cas n'altère en rien le fonctionnement du module PSO. Il ne génère aucun recalcul d'index ni même de modification lourde de la structure (comme c'est le cas, par exemple, pour un algorithme kd-tree qui aurait à régénérer toute l'arborescence dans l'éventualité où le nouveau cas à apprendre n'a pas de cas similaires dans la base de cas. Ce qui est souvent la situation en apprentissage incrémental).
- Prend en charge (en mode natif) les données incomplètes et présentées dans un ordre quelconque puisque l'algorithme intègre la requête dans la fonction objective. Ce qui n'est pas le cas pour la majeure partie des modèles hiérarchisés de base de cas où la requête est représentée à travers les nœuds de l'arborescence. Même si on rencontre dans la littérature des solutions à ce problème<sup>18</sup>, sa prise en charge est souvent prohibitive.

Notre approche s'appuie sur le principe de mémoire accessible par le contenu comme les CRNs. En outre, elle évite l'inconvénient des approches classiques basées indexe. Elle évite également la nécessité du calcul préalable et la construction de liens de similarité qui ajoutent à la complexité de la construction de modèles de mémoire comme c'est le cas pour Fish & Shrink et les CRNs.

Enfin, notre modèle offre une flexibilité (pas de liens préétablis) et une facilité de construction, avec une représentation des connaissances uniforme selon le module PSO. L'objectif visé est donc largement atteint pour cette étape.

La mémoire de cas idéale est celle qui simultanément augmente la vitesse du rappel et améliore la précision et la robustesse de la tâche accomplie par le

---

<sup>18</sup> Ainsi, on trouve la notion de réseaux redondants à caractéristiques partagées (Kolodner 1993) qui créent une variante de réseaux pour chaque variante de requête avec un ordre différent des valeurs d'attributs. De même, dans (Wess, Althoff, et Derwand 1994), les auteurs parlent de l'extension partielle des k-d trees en n-ary trees pour la prise en charge de données incomplètes et non ordonnées.

raisonneur, en particulier l'étape de réutilisation (Bichindaritz, 2008). Comme une extension possible, nous avons ajouté la dimension RGR à la fonction objective. Ainsi, la fonction à optimiser par PSO exprime une sémantique de réutilisation. Nous pouvons considérer que cet objectif a aussi été atteint. Pour la problématique de recherche du bon cas, on retrouve dans la littérature une approche qui aborde cette question à travers la notion de généralisation (Dufour-Lussier et al. 2011). Il y est question d'utiliser une ontologie du domaine afin de généraliser progressivement le cas cible ensuite adapter les cas sources conformément au cas cible généralisé. L'aspect généralisation sur lequel se base cette approche, bien que présent en CBR, pourrait passer à côté de la richesse de description dont disposent les cas histoires. Notre approche ne présente pas un tel risque puisqu'elle ne procède à aucune compilation du cas et l'utilise dans sa forme brute.

Notons qu'au-delà des objectifs initialement fixés, l'analyse de la complexité de l'algorithme de recherche en mémoire de cas était de l'ordre de  $O(T_{max} * M * N)$ , où  $M$  est le nombre de particules de l'essaim,  $N$  est le nombre de dimensions de l'espace problème et  $T_{max}$  est le nombre maximum d'itérations. Les facteurs desquels dépend le coût comprennent  $T_{max}$  et  $M$  qui sont inhérents à l'algorithme, et  $N$  qui est inhérent au problème. Le fait d'avoir le contrôle sur deux paramètres parmi 3 rend cette approche de recherche en mémoire plus qu'intéressante comparativement à la recherche linéaire qui est de l'ordre  $O(Taille)$ , Taille étant le nombre de cas dans la base de cas. C'est un paramètre hors de contrôle. En comparaison, pour une recherche Fish & Shrink, la complexité computationnelle dépend de la connectivité des aspects et de la taille de la requête; aucun des paramètres ne peut être contrôlé. De même pour un modèle aussi performant que les CRN, dont la complexité dépend de la taille de la requête, de la connectivité des entités d'information et de leur spécificité, aucun de ces trois paramètres n'est sous contrôle.

Par ailleurs, Hammond (1989) considérait le CBP comme une nouvelle théorie de la planification dans la mesure où la perception des problèmes de planification comme problèmes de remémoration altère bon nombre de caractéristiques de base des théories de planification, à savoir :

- Au lieu de planifier pour des buts individuels et ensuite fusionner le résultat, le CBP cherche en mémoire un plan qui puisse satisfaire un maximum de ces buts à la fois.
- Plutôt que de récupérer après un échec de planification et oublier ensuite cet échec, le CBP traite ces échecs comme opportunités d'apprentissage sur le domaine.
- Le plan une fois construit ne sera pas écarté après utilisation, il sera sauvegardé dans la base pour des réutilisations futures.

Planifier et apprendre revient donc à réutiliser tout en planifiant une réutilisation future à travers l'apprentissage, le tout dans un même cycle. Il s'agit là d'une réelle économie de la connaissance dont nous avons pu faire l'expérience avec succès à travers cette thèse.

### 7.3 Retour sur les contributions

Les contributions sont décrites en termes de mise en œuvre des solutions aux problèmes dégagés lors de la phase d'analyse :

- Le développement d'un algorithme PSO pour la prise en charge de données mixtes. Même s'il ne faisait pas partie des énoncés de thèse avancés au départ, le besoin du traitement des données mixtes s'est fait sentir tôt dans le processus d'analyse. Notre contribution à ce niveau a été de proposer un algorithme PSO capable de fonctionner avec une description de problème comportant des dimensions de nature mixte et tester son efficacité par rapport à

d'autres approches. Le fait d'avoir pu y répondre par une approche uniforme et non seulement utilisable pour toutes sortes de tâches en PSO, mais aussi extensible à d'autres techniques en classification, fait de cet apport une contribution majeure dans le traitement des données mixtes.

- La mise en œuvre d'un modèle inédit de mémoire basé PSO. Il s'agit là d'une vision révolutionnaire de la mémoire de cas : pas de biais de structuration, pas de complexité de construction, accessibilité par le contenu, flexibilité de représentation, mais aussi facilité d'intégrer d'une part le mécanisme d'interprétation pour la prise en charge de données mixtes et d'autre part la sémantique de la recherche guidée par la réutilisabilité.

- L'intégration du critère d'adaptabilité dans le modèle de mémoire basé PSO constitue une avancée importante dans le domaine de la recherche en mémoire de cas. Notre contribution à ce niveau a été de raffiner le modèle précédent par l'ajout du critère d'adaptabilité comme garant de la qualité du processus de rappel en CBR. La fonction objective associée à l'algorithme PSO a été reformulée afin de supporter la sémantique RGR.

- Au-delà des aspects techniques attrayants de ce modèle, ce dernier présente l'avantage de s'appuyer sur un parcours stochastique de l'espace de recherche en utilisant un mécanisme d'interprétation basé sur les fréquences. Cette particularité revient à éviter ce que Stanovich (2009) avait qualifié de 'mindware gap'. Ainsi, on ne va pas retrouver les cas les plus similaires à la situation uniquement, mais aussi les plus probables. Ce qui rend le modèle plus vraisemblable en combinant les avantages du fast et du slow thinking.

- La modélisation CBR pour la planification constitue une autre contribution dans la mise en place de modèles de connaissance nécessaires à la capture de l'expertise de génération de plan, incluant le modèle de connaissances

de similarité et d'adaptabilité. Le modèle ainsi défini se caractérise par une généralité qui permet de rester indépendant du domaine d'application (et de sa complexité), tout en s'intéressant à la tâche de résolution de problème qu'est la génération de plan. Secondairement, cette généralité garantit la réutilisabilité du modèle.

#### 7.4 Perspectives

Dès le départ nous avons le souci de la généralité à chacun des choix conceptuels effectués, de manière à permettre la réutilisabilité du modèle d'autres domaines d'application tels que le routage dans les réseaux, la robotique...

Quant au modèle de mémoire proposé, nous avons pu démontrer son potentiel de diversification en l'implémentant dans un cadre de conception de menus diététiques. Une perspective plus intéressante serait de considérer l'algorithme de recherche mis au point dans un cadre plus large de recherche d'information ou, plus précisément, dans le cadre du web sémantique. Nous envisageons donc de relever le défi lancé par Leake (2011) de considérer « The web as a memory ». Dans ce cas de figure, les entités d'information servant d'élément de description seraient issues des métadonnées du modèle conceptuel.

Une autre perspective envisageable est l'exploitation du mécanisme d'interprétation pour le traitement des données mixtes dans d'autres approches habituellement performantes pour le traitement de données numériques.



## APPENDICE A Classification basée PSO pour les données mixtes

### A.1 La classification basée PSO

À la base, l'algorithme de la PSO reste le même quelque soit la tâche de résolution de problème ciblée. Ce qui fait la différence c'est la sémantique associée à la fonction objective. Dans le cas de la classification, on utilisera l'algorithme décrit en figure II.4. pour une PSO à valeurs continues et celui de vu en figure III.8. pour une PSO valeurs mixtes (la différence entre ces deux algorithmes se situe au niveau de l'introduction d'une étape d'interprétation).

L'apprentissage supervisé s'attaque au problème de trouver une C-partition plausible de classes pour un espace d'entrée, étant donné une C-partition d'un ensemble d'exemples. La PSO étant essentiellement un outil d'optimisation, le problème de la récupération des classes C est exprimé en termes de récupération C positions optimales de particules correspondant aux centroïdes des C classes. L'algorithme effectue un parcours de l'espace de recherche en testant d'adaptation selon une sémantique de distance (resp. similarité); la particule minimisant (resp. maximisant) cette caractéristique sera retenue comme centroïde.

L'approche PSO pour la classification utilise les principales étapes de (De Falco et al. 2006; Abraham et Grosan, 2006). Soient une base de données avec des C classes et les N dimensions de l'espace d'entrée, le problème de classification est considérée comme trouver les meilleures valeurs réelles des coordonnées de centroïdes dans l'espace à N dimensions. Une classe est alors identifiée par son centroïde.

L'idée est de commencer avec un essaim de particules, chacune représentant potentiellement un centroïde de classe, et de manière itérative affiner les positions des particules au cours de l'étape d'apprentissage pour obtenir le meilleur centroïde pour

chaque classe. Les centroïdes potentiels (au départ tout le monde est candidat) sont évalués par la fonction d'adaptation (équation 7) par rapport à des prototypes de classe et les meilleurs sont retenus pour l'itération suivante. Durant la phase de décision, les centroïdes sont évalués par la fonction d'adaptation par rapport à des prototypes de classe afin d'établir la précision de reconnaissance

$$\psi_i^c(t) = \frac{1}{D_{Train}} \sum_{k=1}^{D_{Train}} d(y_k^c, x_i^c(t)) \quad (7)$$

Où  $y_k^c$  est le  $k^{ième}$  exemplaire de la classe  $c$  dans l'ensemble d'apprentissage de taille  $D_{Train}$  et  $x_i^c(t)$  est la position courante du centroid de la classe  $c$  déterminé par la  $i^{ème}$  particule, les deux vecteurs étant de dimension  $N$  (l'indice de la dimension ayant été omis pour alléger la notation).

Lors du calcul de distance, chaque composant à  $N$  dimensions est divisé par son rang, et la somme des distances des composants est divisée par  $N$ . ainsi, n'importe quelle distance sera normalisée dans l'intervalle  $[0.0, 1.0]$  et il en sera de même pour la fonction objective

Compte tenu de la forme de la fonction objective, le problème devient un problème de minimisation typique.

## A.2 Résultats

L'objectif de cette partie est de présenter plus en détail les résultats de la classification pour les données mixtes. Nous avons utilisé trois bases de données extraites de UCI Repository (Asuncion et Newman, 2007).

Nos résultats ont été obtenus par deux approches:

- Un algorithme de classification PSO en utilisant l'équation (3) pour la mise à jour des positions (appelé A).

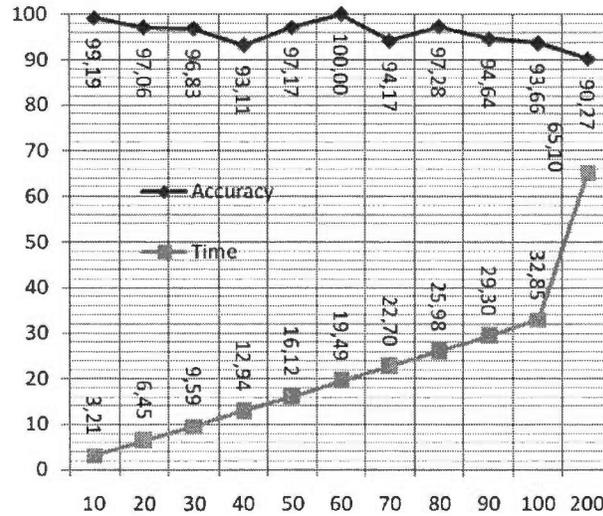
● Un algorithme de classification PSO basée sur la dispersion par le vent et le confinement d'intervalle, en utilisant l'équation (3c) de mise à jour de la position (appelé B).

Tous les résultats qui sont présentés dans cette section représentent la moyenne de 10 essais. Le temps de d'apprentissage est utilisé pour une estimation de la complexité de l'algorithme. Notons que la procédure de décision est basée sur un seuil de tolérance cognitif associé au domaine d'application. Le seuil est la valeur selon laquelle la fonction fitness calculée est considérée comme satisfaisante pour la reconnaissance. Ainsi, c'est la marge d'erreur acceptée lors de la phase de reconnaissance. Comme nous n'avons de directives de choix, nous avons essayé différentes valeurs.

#### A.2.1 Base de données Adult

Comme première série d'expérimentations, nous avons divisé l'ensemble de données en ensemble d'apprentissage et ensemble de test comprenant 2/3 et 1/3 des données (apprentissage = 32561 instances, test = 16281 instances).

Les paramètres PSO ont été réglés à  $v_{max}=0.5$ ,  $v_{min}=-0.5$ ,  $c_1=2.0$ ,  $c_2=2.0$ ,  $w_{max}=0.9$ ,  $w_{min}=0.4$ ; le seuil de tolérance fixé à 0,003 et nous avons fait varier le nombre de particules ainsi que  $T_{max}$ . L'application de l'approche A conduit à une précision de reconnaissance de  $99,83\% \pm 0,003$  pour une population de 30 particules et d'un temps de calcul de 9,78 heures. L'approche B conduit à une précision parfaite à  $\pm 0,003$  pour une population de 20 particules et un temps d'apprentissage de seulement 6,48 heures. Cela montre que l'ajout de la dispersion et du confinement à l'algorithme PSO a un impact positif sur les performances de classification. Ces résultats ont été obtenus avec seulement 10 itérations, le facteur d'inertie évoluant à grands pas entre  $w_{max} = 0,9$  et  $w_{min} = 0,4$ . Cela s'est traduit par une plus grande dispersion de particules pour une meilleure couverture de l'espace de recherche.



**Figure A.1 Résultats pour la base 'Adult': La précision et le temps d'apprentissage pour la PSO avec dispersion et confinement avec un essaim de 10 particules and  $T_{max}$  variant de 10 à 200 itérations.**

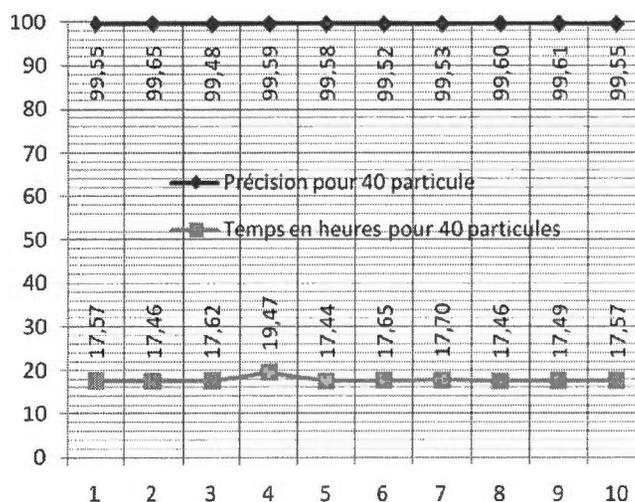
La figure A.1 montre qu'un petit nombre d'itérations réduit le temps d'exécution sans pénaliser la précision de classification. En effet, la solution optimale est atteinte dès le début du cycle d'itération. Par exemple, un score parfait a été atteint en 60 itérations avec seulement un essaim de 10 particules.

Nous remarquons aussi que de bons résultats ont été obtenus avec une taille d'essaim de 30 particules pour l'approche A, tandis que l'approche B a donné de meilleurs résultats en précision et en temps avec seulement 20 particules. Des résultats comparables ont été obtenus en réduisant le nombre de particules à 10 et l'augmentation du nombre d'itérations de 10 à 60.

Face à ces résultats intéressants et afin de s'assurer encore plus que nous ne sommes pas dans une situation de sur-apprentissage, nous avons lancé une autre série d'expérimentations. Nous avons donc effectué une validation croisée, en 10 rotations sur les 48842 instances. Les données ont été réparties en 90% des instances pour l'apprentissage et 10% des instances pour le test, de telle sorte que la fréquence des

classes reste homogène dans l'ensemble des 10 rotations. L'algorithme PSO a été appliqué à dix reprises et nous avons considéré la moyenne des exécutions.

Les paramètres PSO ont été réglés à  $v_{max}=0.5$ ,  $v_{min}=-0.5$ ,  $c_1=1.7$ ,  $c_2=1.7$ ,  $w_{max}=0.9$ ,  $w_{min}=0.4$  et nous avons fait varier le nombre de particules et  $T_{max}$ . Le seuil de tolérance a été fixé à 0,005. Comme l'approche B obtenait de meilleurs résultats dans nos expériences préliminaires, nos résultats de validation croisée ont été obtenus avec cette approche uniquement. La précision de reconnaissance moyenne était de  $89,84 \pm 0,005\%$  pour une population de 10 particules, avec un temps de calcul de 4,45 heures.

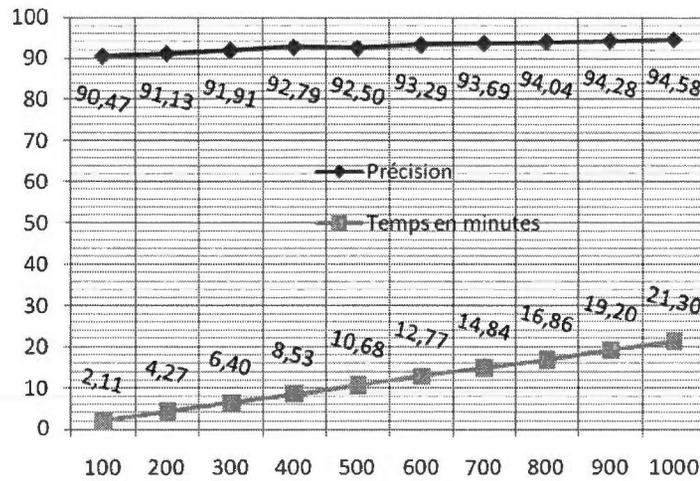


**Figure A.2 Résultats pour la base 'Adult': Précision et temps d'apprentissage pour la PSO avec mécanismes de dispersion et de confinement pour un essaim de 40 particules et 10 rotations (axe des x) en validation croisée.**

Un résultat substantiellement meilleur a été obtenu lorsque la taille de la population de particules a été augmentée à 40 particules (figure A.2). La précision de reconnaissance moyenne est passée à  $99,56\% \pm 0,005$ . Toutefois, le temps de calcul a bondi à 17,74 heures, montrant que dans la pratique, un compromis doit être fait entre la précision et le temps de d'apprentissage, selon le contexte.

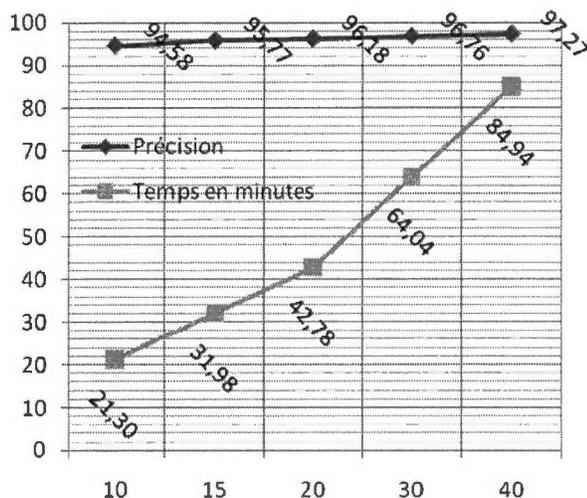
#### A.2.2 La base de données 'CMC'

Nous avons donc effectué une validation croisée en 10 rotations sur les 1473 instances. Les données ont été réparties en 90% des instances pour l'apprentissage et 10% des instances pour le test, de telle sorte que la fréquence des classes reste homogène dans l'ensemble des 10 rotations. L'algorithme PSO a été appliqué à dix reprises et nous avons considéré la moyenne des exécutions. Les paramètres ont été choisis comme suit :  $v_{max}=0.5$ ,  $v_{min}=-0.5$ ,  $c_1=2.0$ ,  $c_2=2.0$ ,  $w_{max}=0.9$ ,  $w_{min}=0.4$  et nous avons fait varier le nombre de particules et  $T_{max}$ . Le seuil de tolérance a été fixé à 0,027. Nos résultats de validation croisée ont été obtenus avec l'approche B.



**Figure A.3 Résultats pour la base CMC: Précision et temps d'apprentissage pour la PSO avec mécanismes de dispersion et de confinement pour un essaim de 10 particules et différents nombres d'itérations (axe des x) en validation croisée (chaque valeur sur la courbe est la moyenne de 10 rotations, chacune d'elles étant la moyenne de 10 essais).**

La figure A.3 montre les résultats de la validation croisée lorsque l'on fait varier le nombre d'itérations. La moyenne de la précision est de 94.58%  $\pm$ 0.027 pour une population de 10 particules, avec un temps d'apprentissage de 22.30 minutes.



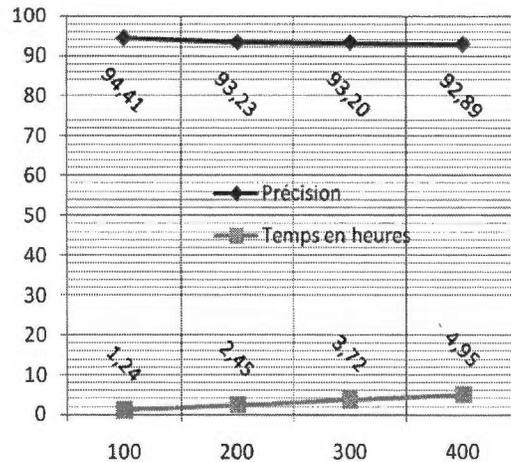
**Figure A.4 Résultats pour la base CMC: Précision et temps d'apprentissage pour la PSO avec mécanismes de dispersion et de confinement pour 1000 itérations et différentes tailles de l'essaim (axe des x) en validation croisée (chaque valeur sur la courbe est la moyenne de 10 rotations, chacune d'elles étant la moyenne de 10 essais).**

Lorsque l'on fait varier la taille de l'essaim (figure A.4), la précision continue à augmenter ainsi que le temps d'apprentissage.

#### A.2.3 La base de données 'Abalone'

Nous avons effectué une validation croisée en 10 rotations sur les 4177 instances. Les données ont été réparties en 90% des instances pour l'apprentissage et 10% des instances pour le test, de telle sorte que la fréquence des classes reste homogène dans l'ensemble des 10 rotations. L'algorithme PSO a été appliqué à dix reprises et nous avons considéré la moyenne des exécutions.

Les paramètres ont été choisis comme suit :  $v_{max}=0.5$ ,  $v_{min}=-0.5$ ,  $c_1=2.0$ ,  $c_2=2.0$ ,  $w_{max}=0.9$ ,  $w_{min}=0.4$  et nous avons fait varier le nombre de particules et  $T_{max}$ . Le seuil de tolérance a été fixé à 0,027. Nos résultats de validation croisée ont été obtenus avec l'approche B.



**Figure A.5 Résultats pour la base 'Abalone': Précision et temps d'apprentissage pour la PSO avec mécanismes de dispersion et de confinement pour un essaim de 10 particules et différents nombres d'itérations (axe des x) en validation croisée (chaque valeur sur la courbe est la moyenne de 10 rotations, chacune d'elles étant la moyenne de 10 essais).**

La figure A.5 montre les résultats de la validation croisée. La moyenne de la précision est  $94.41\% \pm 0.027$  pour une population de 10 particules, avec un temps d'apprentissage de 1.24 heure.

### A.3 Discussion

Une littérature foisonnante fait référence à la base 'Adult' (Kohavi, 1996; Kohavi, Becker et Sommerfield, 1997; Cheng et Greiner, 2001; Hoegaerts et al., 2004; Kao et al., 2004 et Abbasian, 2010). On retrouve différentes approches : les arbres de décision (C4.5), les approches Bayésiennes simples, en arbre ou en réseaux (SBC, NB tree, TAN, BAN et GBN), les machines à vecteurs supports (SVM), les k-plus proches voisins (k-NN) et les réseaux de neurones artificiels (ANN). Notons la difficulté de procéder à la comparaison, dans la mesure où ce ne sont pas tous les travaux revus qui utilisent la totalité de la base de données. Il y en a qui se limite à l'exploitation d'un sous-ensemble des 48842 instances, ou encore un sous ensemble des 14 attributs. De plus, la base 'Adult' présente 7% du nombre total d'instances avec des valeurs manquantes. Certains des travaux revus ignorent dans leur

expérimentation les instances à valeurs manquantes. Dans l'approche PSO, les résultats expérimentaux ont été obtenus en utilisant la totalité des instances et des attributs, y compris les instances comportant des valeurs manquantes. Les résultats de l'approche PSO correspondent à des taux de reconnaissance nettement supérieurs aux autres approches. Par contre les délais d'apprentissage augmentent aussi. Ceci peut être expliqué par la taille importante de la base de données.

De très nombreux articles existent dans la littérature traitant de la classification des données de la base 'CMC' (Lim, Loh et Shih, 2000; Cios et Kurgan, 2004; Rochas, Cortez et Nevez, 2004; Bacardit et Butz, 2007). Ceux-ci utilisent principalement des méthodes statistiques (plus proches voisins, POLYCLASS), arbres de décision (C4.5 renforcée, QUEST) et, plus récemment, des approches hybrides (inductives : CLIP4, connexionnistes : CM ou basée population : PM, DM, LM, BM, GAssist et XCS).

Rappelons que la Base 'CMC' a été considérée comme l'une des plus difficiles à classifier par (Lim, Loh et Shih, 2000). L'approche PSO utilisée l'emporte largement du point de vue de la précision. Néanmoins du point de vue temps d'apprentissage, elle se trouve classée deuxième après les approches hybrides basées population dont la précision est supérieure à 60 %, sachant que ces approches applique un critère d'arrêt au bout de 1000 secondes de temps d'apprentissage.

De multiples papiers existent dans la littérature en relation avec la base de données 'Abalone' (Uysal et Altay, 2004; Ting et Witten, 1999; Snelson, Rasmussen et Ghahramani, 2004). Ceux-ci utilisent principalement les k-plus proches voisins (KNN), les arbres de décision (C4.5), les régressions basées règles (RULE), les régressions basées instances (RFPF) les approches Bayésiennes (NB) et les approches basées sur les processus gaussiens généralisés (GP).

Le travail sur les processus gaussiens généralisés l'emporte en taux de reconnaissance et la PSO est classée en seconde position. Précisons tout de même que

les résultats obtenus par l'approche GP sont sur une proportion train/test de 1000/3177 instances, ce qui rend cette comparaison biaisée.

#### A.4 Analyse de la complexité algorithmique

En dernier lieu, il faut préciser que l'algorithme étant le même, sa complexité reste la même. En l'occurrence elle est de l'ordre de  $O(T_{max} * M * N * Taille)^{19}$ .

Nous pouvons par ailleurs, noter que les facteurs desquels dépend le coût comprennent  $T_{max}$  et  $M$  qui sont inhérents à l'algorithme, et  $N$  et  $Taille$  qui sont inhérents au problème.

En dépit du fait que  $M \ll T_{max}$  et  $N \ll Taille$ , nous ne pouvons négliger  $M$  ou  $N$  puisqu'il s'agit de facteurs multiplicateurs de  $T_{max}$  et  $Taille$ . En conséquence, un compromis doit être envisagé lors du choix des valeurs de  $T_{max}$  et  $M$ , puisque  $N$  et  $Taille$  sont hors de notre contrôle.

La complexité temporelle de la phase d'apprentissage est supérieure à celle de plusieurs techniques dans le domaine. Par exemple, pour l'algorithme ID3 (Steinhaeuser, Chawla et Kogge, 2006). Construire un arbre de décision sur un ensemble de données nominale a une complexité d'ordre  $O(Taille * P)$  et d'ordre  $O(Taille^2 * P)^{20}$  pour les données continues,  $P$  étant la profondeur de l'arbre généré. La complexité en temps de l'algorithme standard C4.5 est d'ordre  $O(Taille * P^2)$ . Notez que là aussi la complexité en temps de C4.5 est supérieure à  $O(Taille * P^2)$  en présence d'attributs numériques (Su et Zhang, 2006). L'analyse de la complexité d'un réseau neuronal est d'ordre  $O(E^3)$ ,  $E$  étant le nombre de neurones (Gelenbe, 1993).

Au stade de l'exploitation, la traversée d'un arbre de décision est certainement plus gloutonne en temps.

---

<sup>19</sup>  $M$  : le nombre de particules de l'essaim,  $N$  : le nombre de dimensions de l'espace problème,  $T_{max}$  : le nombre maximum d'itérations et  $Taille$  : le nombre d'instances de la base de cas.

<sup>20</sup> Dans ce cas la complexité est pire que celle de la PSO, puisque quadratique sur la taille.

## Bibliographie.

- A., Abraham, Grosan, C. (2006). *Swarm Intelligence in Data Mining*. Ramos, V., (Eds.), Springer-Verlag Berlin Heidelberg.
- Aamodt, A., Plaza, E. (1994). Case-Based Reasoning : Foundational Issues. *AI-Communications*, vol.7(1), pp. 39-59.
- Abbasian, H., Drummond, C., Japkowicz N. and Matwin, S. (2010). Robustness of Classifiers to Changing Environments. *AI'10: Advances in Artificial Intelligence, LNCS*, vol. 6085/2010, pp. 232-243.
- Aha, D.W. (1998). The Omnipresence of Case-Based Reasoning in Science and Application. *Knowledge-Based Systems journal*, vol. 11(5-6), pp. 261-273.
- Al-kazemi B, Mohan CK. (2000). Multi-phase discrete particle swarm optimization. *Proc. Fourth international workshop on frontiers in evolutionary algorithms, Kinsale, Ireland*, pp. 622-625, April 2002.
- Allen, B. (1994). Case based reasoning : business applications. *C-ACM*, vol37(3), pp. 40-42, March 1994.
- Angeline, P.J. (1998). Using selection to improve particle swarm optimization. *Proc. IEEE congress on evolutionary computation*, pp. 84-89, Anchorage, Alaska, May 1998.
- Ashley, K. D., Rissland, E. L. (1988). A Case Based Approach to Modelling legal expertise. *IEEE Expert*, vol. 3(3), pp.70-77.

- Asuncion, A. & Newman, D.J. (2007). UCI Machine Learning Repository [<http://www.ics.uci.edu/~mllearn/MLRepository.html>]. Irvine, CA: University of California, School of Information and Computer Science.
- Bacardit, J., Butz, M. V. (2007). Data Mining in Learning Classifier Systems: Comparing XCS with GAssist. In *Learning Classifier Systems, LNCS*, , Vol. 4399/2007, pp. 282-290, Springer-Verlag Berlin Heidelberg.
- Bartsch-Spörl B. & al. (1999). Case-Based Reasoning Surveys and Future Direction. *XPS-99: Knowledge-Based Systems. Survey and Future Directions, LNCS*, vol. 1570, pp. 67-89, March 1999, Springer-Verlag Berlin Heidelberg.
- Beni, G. (2005a). Order by disordered actions in Swarms. *Swarm Robotics WS 2004: LNCS*, vol. 3342/2005, pp. 153–171, Springer-Verlag Berlin Heidelberg.
- Beni, G. (2005b). From Swarm Intelligence to Swarm Robotics. E. Sahin and W.M. Spears (Eds.): *Swarm Robotics WS 2004: LNCS*, vol. 3342/2005, pp. 1–9, Springer-Verlag Berlin Heidelberg.
- Beni, G. Wang, J. (1989). Swarm Intelligence in Cellular Robotic Systems. *Proc. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, June 26–30, 1989 [s.p.]*.
- Bergmann R., Wilke W. (1998). Towards a new Formal Model of Transformational Adaptation in Case Based Reasoning. *Proc. of ECAI*, po.53-57, ed. John Wiley and sons, 1998, [s.l.].
- Bergmann, R. , Munoz-Avila, H., Veloso, M. and Melis, E. (1998). Case-based reasoning applied to planning tasks. In Mario Lenz, Brigitte Bartsch-Spörl, H.-D. Burkhard, and Stefan Wess, editors, *Case-Based Reasoning Technology from Foundations to Applications, 1998, Springer-Verlag, [s.p.]*.

- Bergmann, R., Althoff, K. (1998). Methodology for Building Case-Based Reasoning Applications. *LNAI, vol. 1400/1998, Springer-Verlag Berlin Heidelberg [s.p.]*.
- Bichindaritz, I. (2008). Memory Structures and Organization in Case-Based Reasoning. *Studies in Computational Intelligence (SCI) 73, Ch. 6, pp. 175–194, Springer-Verlag Berlin Heidelberg*.
- Bichindaritz, I. (2006). Memory Organization As The Missing Link Between Case-Based Reasoning And Information Retrieval In Biomedicine. *Computational Intelligence, vol. 22( 3/4), pp. 148-160, Blackwell Publishing*.
- Bonabeau, E., Dorigo, M., Theraulaz, G. (1999). Swarm Intelligence. From Natural to Artificial systems. Ed. Oxford University Press.
- Boriah, S., Chandola, V., Kumar, V. (2008). Similarity Measures for Categorical Data: A Comparative Evaluation. *Proc. SIAM Data Mining Conference, Atlanta, GA, pp. 243-254, April 2008*.
- Brosset. D. (2008). Description D'itinéraire En Milieu Naturel : Modèle Intégré De Description Verbale Et De Représentation Spatiale Au Sein Des SIG. Thèse de doctorat de de l'École Nationale Supérieure d'Arts et Métiers en Informatique, Paris, France.
- Brown, M. (1994). A Memory Model for Case Retrieval by Activation Passing. PhD Dissertation, Manchester University, United Kingdom.
- Caplat, G. (2002). Modélisation cognitive et résolution de problème. Ed. Presses Polytechniques et Universitaires Romandes, Lausanne.
- Carrillo-Ramos, A. C. (2007). Agents Ubiquitaires Pour Un Accès Adapté Aux Systèmes D'information :Le Framework Pumas. Thèse De Doctrat De L'universite Joseph Fourier En Informatique,2007.

- Chandramouli, K. and Izquierdo, E. (2006). Image Classification using Chaotic Particle Swarm Optimization. *Proc. International Conference on Image Processing (ICIP '06), Atlanta, GA, pp. 3001-3004, 8-11 Oct. 2006.*
- Cheng, J., Greiner, R. (2001). Learning Bayesian Belief Network Classifiers: Algorithms and System. *Proc. 14<sup>th</sup> Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence, LNCS, vol. 2056/2001, Pages: 141 – 151 [s.l.].*
- Cios, K.J., Kurgan, L.A. (2004). CLIP4: Hybrid inductive machine learning algorithm that generates inequality rules. *Information Sciences in Soft Computing Data Mining journal, vol. 163(1-3), pp : 37–83, Elsevier.*
- Clerc, M. (2005). L'optimisation par essaim particulaire: versions paramétriques et adaptatives. Ed. Hermes science publications, Lavoisier, Paris.
- Consoli, S., Moreno-Perez , J. A., Darby-Dowman, K., Mladenovic, N. (2009). Discrete Particle Swarm Optimization for the minimum labelling Steiner tree problem. *In Nat Comput. DOI 10.1007/s11047-009-9137-9, ed. Springer Science+Business Media B.V. 2009.*
- DeFalco, I., Della Cioppa, A., Tarantino, E. (2006). Evaluation of Particle Swarm Optimization Effectiveness in Classification. *LNAI, vol. 3849/2006, pp. 164-171, Springer-Verlag Berlin Heidelberg.*
- Diday E. et al. (1982). *Eléments d'analyse de données.* Éd. Dunod.
- Dorigo, M. Socha, K. (2007). An Introduction to Ant Colony Optimization. T. F. Gonzalez, *Approximation Algorithms and Metaheuristics, CRC Press [s.p.].*
- Dorigo, M. (2007). Editorial of Swarm Intelligence Journal, In *Swarm Intelligence Journal, voll(1), Springer New York [s.p.].*

- Dorigo, M., Birattari, M., Stützle, T. (2006). Ant Colony Optimization : Artificial Ants as a Computational Intelligence Technique. *IEEE Computational Intelligence Magazine*, 2006 [s.p.].
- Dorigo, M., Maniezzo, V. and Colomi, A. (1996). Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, vol. 26(1), pp. 29-41.
- Dufour-Lussier, V., Lieber, J., Nauer, E., Toussaint, Y. (2011). Improving Case Retrieval by Enrichment of the Domain Ontology. *ICCBR 2011: LNAI*, vol. 6880/2011, pp. 62–76, Springer-Verlag Berlin Heidelberg.
- Engelbrecht, A. P. (2007). Computational Intelligence, An Introduction. John Willey & Sons Editions.
- Erzan M. et al. (1999). Réutilisation logicielle: Guide pratique et retour d'expérience. ed. Eyrolles, 1999.
- Farreny, H., Ghallab, M. (1987). Éléments d'intelligence artificielle, Édition Hermès, Paris.
- Fikes, R. E. & Nilsson, N. J. (1971). STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence*. Vol. 2(3–4). pp. 189–208.
- Fuch, B. (1997). Représentation des connaissances pour le raisonnement à partir de cas. Système Rocode. Thèse de Doctrat de l'université Jean Monnet de Saint\_Etienne, France.
- Garnier, J. Gautrais, Theraulaz, G. (2007). The Biological Principles of Swarm Intelligence. *In Swarm Intelligence Journal*, vol. 1(1), pp. 3-31, Springer New York.

- Gelenbe, E. (1993). Learning in the Recurrent Random Neural Network. *Neural Computation*, vol. 5(1). Pp. 154–511.
- Gutjahr, W. J. (2007). Mathematical runtime analysis of ACO algorithms: survey of an emerging issue. *Swarm Intelligence Journal*, vol. 1(1), pp. 59-79, Springer New York.
- Hammond, K.J. (1989). Case-Based Planning: Viewing Planning as a Memory Task. Academic Press, Boston.
- Hassan, H. A., Yassin, I. M. Halim, A. K., Zabidi, A., Majid, Z. A., Abidin, H. Z. (2009). Logical Effort using a Novel Discrete Particle Swarm Optimization Algorithm. *Proc. 5<sup>th</sup> International Colloquium on Signal Processing & Its Applications (CSPA)*, pp. 432-438, 6-8 March [s.l.]
- Haton, J.-P. (1993). Le raisonnement dans les systèmes à base de connaissances. *Le courrier du CNRS*, n° 80, 1993 [s.p.].
- Higashi N, Iba H. (2003). Particle swarm optimization with Gaussian mutation. *Proc. of the IEEE swarm intelligence symposium 2003 (SIS 2003)*, Indianapolis, Indiana, USA, pp 72–79, 24-26 March 2003.
- Hoegaerts, L., Suykens, J., Vandewalle, J., De Moor, B. (2004). Primal Space Sparse Kernel Partial Least Squares Regression for Large Scale Problems. *Proc. International Joint Conference on Neural Networks (IJCNN04)*, vol. 1, pages 561–566, 25-29 July 2004.
- Holden, N, Freitas, A. (2005). A HYBRID PARTICLE SWARM/ANT COLONY ALGORITHM FOR THE CLASSIFICATION OF HIERARCHICAL BIOLOGICAL DATA. *Proc. IEEE Swarm Intelligence Symposium, pasadena, CA*, pp. 100-107, 8-10 june 2005.

- Kahneman, D., Tversky, A. (1973). On the psychology of prediction. *In Psychological Review*. Vol. 80(4), pp. 237-251, July 1973.
- Kao Wei-Chun, Kai-Min Chung, L. Assun, Chih-Jen Lin. (2004). Decomposition Methods for Linear Support Vector Machines. *In Neural Computation*, vol. 16(8), pp. 1689 – 1704, August 2004.
- Kennedy J, Eberhart R. (1995). Particle swarm optimization. *Proc. 4<sup>th</sup> IEEE international conference on neural networks, Perth, Australia*, pp. 1942–1948, 1995.
- Kennedy J, Eberhart R. (1997). A discrete binary version of the particle swarm algorithm. *Proc. IEEE conference on systems, man, and cybernetics*, vol 5, pp 4104–4108.
- Kennedy, J., Eberhart, R. (2001). *Swarm Intelligence*. Ed. Morgan Kaufmann.
- Kodratoff, Y. (1993). L'apprentissage. *Le Courrier du CNRS N°80, Février 1993* [s.p.].
- Koehler, J. (1994). Avoiding Pitfalls in Case-based Planning. *Proc. AIPS 1994*, pp. 104-109, *AAAI 1994* [s.l.n.é.].
- Kohavi, R. (1996). Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid. *Proc. 2<sup>nd</sup> International Conference on Knowledge Discovery and Data Mining (KDD-96)*. p.202-207. *AAAI Press* [s.l.].
- Kohavi, R., Becker, B., Sommerfield, D. (1997). Improving Simple Bayes. *Proc. 9<sup>th</sup> European Conference on Machine Learning*. p. 78-87. *Springer Verlag, Heidelberg* [s.l.].

- Kolodner J. (1991). Improving human decision through case based decision aiding. *AI magazine*[s.p.].
- Kolodner J., (1992). Judging which is the best case for a case based reasoner. *DARPA Workshop on CBR* [s.p.].
- Kolodner J. (1993). Case Based Reasoning. Éd. Morgan Kaufmann.
- Krink T, Løvbjerg M. (2002). The lifecycle model: combining particle swarm optimization, genetic algorithms and hillclimbers. *Proc. of parallel problem solving from nature VII (PPSN 2002)*. LNCS, vol. 2439/2002, pp 621–630, Springer.
- Leake, D. (2011). Assembling Latent Cases from the Web: A Challenge Problem for Cognitive CBR. *ICCBR-2011: Workshop on Human-Centered and Cognitive approaches to CBR*, Greenwich, UK, 2011. 7 pages.
- Lenz, M. et al. (1998). *Diagnosis and decision support*”, LNAI, vol. 1400, Springer [s.p.].
- Lenz, M. (1999). Case Retrieval Nets as a Model for Building Flexible Information Systems. PhD Dissertation, Humboldt University, Berlin, Germany.
- Lim, T.-S., Loh, W.-Y., Shih, Y.-S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, vol. 40, pp.203–228 [s.l.n.é.].
- Liu H., Abraham A. (2005). Fuzzy adaptive turbulent particle swarm optimization. *Proc. 5<sup>th</sup> international conference on hybrid intelligent systems (HIS'05)*, Rio de Janeiro, Brazil, 6–9 November 2005 [s.p.].

- Lopez de Mantara R. et al. (2005). Retrieval, reuse, revision, and retention in case based Reasoning. *The Knowledge Engineering Review*, vol. 20(3), pp. 215-240, Cambridge University Press.
- Machroum, E. P., (2007). Information verbale et information graphique pour la description d'itinéraires : une approche pluridisciplinaire. Thèse de doctorat de l'université Paris 5 en sciences cognitives, Paris, France.
- Malek, M. (1997), Un Modèle Hybride de Mémoire Pour le Raisonnement à Partir de Cas. Thèse de Doctrat de l'université Joseph Fourier, Grenoble, France, 1997.
- McGinty, L. and Smyth, B. (2001). Collaborative Case Based Reasoning: Applications in Personalised Route Planning. *Proc. International Conference on Case Based Reasoning (ICCBR) Vancouver, Canada, pp.362-376, August 2001.*
- Mikki S. M. and Kishk, A. A. (2008). Particle Swarm Optimizaton: A Physics-Based Approach. *Synthesis Lectures in Computational Electromagnetics series, Morgan & Claypool Publishers.*
- Mokhtari, A. , Pivert, O., Hadjali, A. (2009). Integrating Complex User Preferences into a Route Planner: A Fuzzy-Set-Based Approach. *Proc. IFSA-EUSFLAT 2009, Lisbon, Portugal, July 20-24, pp. 501-506 [s.l.]..*
- Moreno-Pérez JA, Castro-Gutiérrez JP, Martinez-Garcia FJ, Melian B, Moreno-Vega JM, Ramos J. (2007). Discrete Particle Swarm Optimization for the p-median problem. *Proc. 7th metaheuristics international conference, Montréal, Canada [s.p.]*.
- Nebel, N. and Koehler, J. (1995). Plan reuse versus plan generation: a theoretical and empirical analysis. *Artificial Intelligence*, vol. 76, p. 427–454.

- Newell, A. & Simon, H. A. (1972). *Human Problem Solving*. Englewood Cliffs, NJ: Prentice-Hall [s.p.].
- Norman, W. (2006). *A Survey of the Applications of Swarm Intelligence*. Internal Report, Clemson University, 2006.
- Omran, M. G. H., Engelbrecht, A. and Salman, A. (2007). Barebones particle swarm for integer programming problems. *Proc. IEEE Swarm Intelligence Symposium*, pp. 170-175, 1-5 April 2007 [s.l.].
- Omran, M.G.H., Engelbrecht, A.P, and Salman, A. (2005). Dynamic Clustering using Particle Swarm Optimization with Application in Unsupervised Image Classification. *In Enformatica Transactions on Engineering, Computing and Technology*, vol. 9, pp. 199-204, Nov. 2005. ISSN 1305-5313
- Pampara G, Franken N, Engelbrecht AP. (2005). Combining Particle Swarm Optimisation with angle modulation to solve binary problems. *Proc. IEEE congress on evolutionary computing, Edinburgh, Scotland*, vol 1, pp. 89–96.
- Petalas, Y.G., Parsopoulos, K.E., Vrahatis, M.N. (2007). Memetic particle swarm optimization. *Annals of Operational Research*, vol. 156, pp. 99–127. DOI 10.1007/s10479-007-0224-y
- Pugh, J Martinoli, A. (2006). Discrete multi-valued particle swarm optimization. *Proc. IEEE swarm intelligence symposium, Indianapolis, IN*, vol 1, pp 103–110, May 12-14, 2006.
- Régner, P. (1992). *Recherche et exploitation du parallélisme en planification*. Thèse de doctorat de l'université Paul Sabatier en Intelligence Artificielle, Toulouse, France.
- Régner, P., Vidal, V. (2004). *Algorithmique de la planification en IA*. Éd. Cépaduès.

- Richter, M. (2004). Similarity Based Retrieval. Research Report, University of Kaiserslautern, Germany, 2004 [s.l.n.é.].
- Rippol, T., Tricot, A. (1995). Quelques points de repères sur l'évolution de l'étude du raisonnement en psychologie cognitive. *Cahiers pédagogiques*, vol. 344-345, N°spécial « apprendre à raisonner ? », pp. 37-40, 1995.
- Rissland, E., Skalak, D. (1989). Case Based Reasoning in a Rule Governed domain. *Proc. 5th Conference on AI Applications, Miami, FL*, pp. 45-53, 6-10 Mar 1989.
- Rochas, M., Cortez, P., Nevez J. (2004). Evolutionary neural network learning algorithms for changing environments. *WSEAS Transactions on Systems*, vol. 3:2, p. 596-601, ISSN 1109-2777.
- Rougegrez-Loriette, S. (1994). Prédiction de processus à partir de comportement observés : le système REBECAS. Thèse de doctorat d'université, Institut Blaise Pascal Paris VI, LAFORIA TH94/05, Juillet 1994.
- Sahin, E. (2005). Swarm Robotics : From Sources of inspiration to domains of application. *E. Sahin and W.M. Spears (Eds.): Swarm Robotics WS 2004, LNCS*, vol. 3342/2005, pp. 10-20, Springer-Verlag Berlin Heidelberg.
- Schaaf, J. W. (1996). Fish and Shrink. A next step towards efficient case retrieval in large scaled case bases. *Advances in Case-Based Reasoning, LNCS*, vol. 1168/1996, p. 362-376, Springer Berlin / Heidelberg.
- Site web des systèmes de transport intelligents du ministère des transports du Canada, url : <http://www.its-sti.gc.ca>.
- Smail, M. (1994). Raisonnement à base de cas pour une recherche évolutive d'information; Prototype Cabri-n. Vers la définition d'un cadre d'acquisition de

- connaissances. Thèse de doctorat d'université, Univ. Henri Pointcaré — Nancy I, Octobre 1994.
- Smyth, B., Keane, M. T. (1999). Adaptation-Guided Retrieval: Questioning the Similarity Assumption in Reasoning. *Journal of Artificial Intelligence*, vol. 102(2), pp. 249-293, Elsevier .
- Smyth, B., McKenna, E. (1998). Modelling the competence of case bases. *LNAI*, vol. 1488/1998, pp. 208-220, Springer 1998.
- Snelson, E., Rasmussen, C.E., and Ghahramani, Z. (2004). Warped Gaussian Processes. *NIPS 2003: Advances in Neural Information Processing Systems 16*, pp. 337-344 Cambridge, MA: MIT Press [s.l.].
- Spalazzi, L. (2001). A Survey on Case-Based Planning. *Artificial Intelligence Review* vol. 16, pp. 3–36, Kluwer Academic Publishers, Netherlands.
- Stanovich, K. (2009). Rational and irrational thought: the thinking that IQ tests miss. *In scientific American Mind*, p. 34-39, November/December 2009.
- Steinhaeuser, K., Chawlam N. V. and Kogge, P. M. (2006). Exploiting Thread-Level Parallelism to Build Decision Trees. *ECML/PKDD Workshop on Parallel Data Mining (PDM)*, Berlin, Germany [s.p.].
- Su, J., Zhang, H. (2006). A Fast Decision Tree Learning Algorithm. *Proc. 21st National Conference on Artificial Intelligence*, Boston, MA, Vol. 1, pp. 500-505, July 16-20].
- Ting, K., M., Witten, I. H. (1999). Issues in stacked generalization. *Journal of Artificial Intelligence Research*, vol. 10, pp.271-289. Morgan Kauffmann Publishers.

- Uysal L. And Altay Gu H. (2004). Instance-Based Regression by Partitioning Feature Projections. *Applied Intelligence journal*, vol.21, pp. 57–79, Kluwer Academic Publishers.
- Van den Bergh F, Engelbrecht A.P. (2004). A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, vol. 8(3), pp.225–239, June 2004.
- Watson I. (1997). Applying Case-Based Reasoning: techniques for enterprise systems. Éditions Morgan Kaufman.
- Wess, S., Althoff, K.-D. & Derwand, G. (1994). Using k-d Trees to Improve the Retrieval Step in Case-Based Reasoning. *Proc. EWCBR'93*, In: S. Wess, K. -D. Althoff & M. M. Richter (eds.), *Topics in Case-Based Reasoning*, pp. 167-181, Springer-Verlag [s.p.].
- Wiltgen, B., Ashok K. Goel, A. K., Vattam, S. (2011). Representation, Indexing, and Retrieval of Biological Cases for Biologically Inspired Design. *Proc. ICCBR 2011: LNAI*, vol. 6880/2011, pp. 334–347, Springer-Verlag Berlin Heidelberg.
- Yang S, Wang M, Jiao L. (2004). A Quantum Particle Swarm Optimization. *Proc. of CEC2004, the congress on evolutionary computing*, vol 1, p 320–324 [s.l.].
- Yin Tan. (2010). Particle Swarm optimization algorithms Inspired by Immunity-Clonal Mechanism and Their applications to Spam detection. *International Journal of Swarm Intelligence Research*, vol. 1(1), pp. 64-86, January-March 2010.
- Zhang WJ, Xie X.F. (2003). DEPSO: hybrid particle swarm with differential evolution operator. *IEEE interenational conference on systems, man and cybernetics (SMCC)*, Washington DC, USA, pp. 3816–3821, 5-8 Oct. 2003.