



Université Larbi Tébessi – Tébessa  
Faculté des Sciences Exactes, Science  
de la Nature et de la Vie

Département des Mathématiques et Informatique



LAMIS

Laboratoire de Mathématiques,  
Informatique et Systèmes

THÈSE

Pour obtenir le grade de docteur  
3<sup>ème</sup> cycle L.M.D. en informatique  
Option : Systèmes d'Information Coopératifs

Thème

**Systeme intelligent d'aide à la décision  
pour la planification du projet urbain**

Présentée par : **KHELIFA Boudjemaa**

Soutenue le : 24/03/2021, devant le jury composé de :

Dr. AMROUNE Mohamed (MCA), *Université Larbi Tébessi, Tébessa,*

Président ;

Pr. LAOUAR Mohamed Ridda (Professeur), *Université Larbi Tébessi, Tébessa,*

Directeur de thèse ;

Pr. KAZAR Okba (Professeur), *Université Mohamed Khider, Biskra,*

Examineur ;

Dr. MESSAOUD Abbas (MCA), *Université Echahid Hamma Lakhdar, El Oued,*

Examineur.

Année universitaire : 2020/2021

# **Système intelligent d'aide à la décision pour la planification du projet urbain**

**KHELIFA Boudjema**  
boudjema.khelifa@univ-tebessa.dz

**Laboratoire de Mathématiques, Informatique et Systèmes (LAMIS),  
Université Larbi Tébessi - Tébessa, Algérie.**



**Laboratoire de Mathématiques,  
Informatique et Systèmes**



*À mes parents ; À ma famille ;  
À mes amis ; À tous ceux qui m'ont  
enseigné*

# Remerciements

*Je tiens tout d'abord à remercier mon directeur de thèse Mr. LAOUAR Mohamed Ridda, Professeur à l'Université Larbi Tébessi - Tébessa, qui m'a guidé pendant plusieurs années. Je le remercie pour toute l'attention qu'il a portée à mon travail, pour sa confiance ainsi que son soutien moral. Sans lui ce travail n'aurait pas pu voir le jour.*

*Je tiens à remercier sincèrement Mr. AMROUNE Mohamed, membre du laboratoire LAMIS et MCA à l'Université Larbi Tébessi - Tébessa, d'avoir accepté de lire et de juger ce travail et d'avoir su se rendre disponible lors de nos échanges scientifiques.*

*Que Mr. KAZAR Okba, Professeur à l'Université Mohamed Khider - Biskra, et Mr. MESSAOUD Abbas MCA à l'Université Echahid Hamma Lakhdar - El Oued, reçoivent ma sincère sollicitude pour avoir bien voulu faire partie de ce jury, avec qui c'est un privilège de pouvoir partager mon travail.*

*J'exprime également ma vive reconnaissance à Mr. LAURINI Robert, Professeur à l'Institut des Systèmes de Connaissances « Knowledge Systems Institute », Illinois, États-Unis d'Amérique, pour m'avoir fait l'honneur de relire ma thèse et pour ses conseils qui ont permis d'enrichir ce travail.*

*Je remercie tous mes collègues et amis du laboratoire LAMIS pour l'aide qu'ils m'ont apportée.*

*Je tiens aussi à remercier ceux qui, dans des périodes plus difficiles, ont été là pour lire mon travail et m'aider à avancer. Particulièrement mon cher ami Mourad Soualmia.*

*Enfin ma famille, sans qui cela n'aurait pas été possible. En particulier mes parents, ma compagne, et mes enfants qui ont été mon ultime motivation.*



## Résumé :

Ces derniers temps, le milieu urbain a connu une expansion extraordinaire ce qui multiplie les problèmes urbains de jour en jour et provoquant un phénomène urbain plus dynamique et plus complexe. En raison des aspirations des décideurs et du peu de temps alloué, la planification de projets urbains demeure une tâche fastidieuse, qui mène souvent à des choix arbitraires pour trouver de bonnes adéquations des projets en fonction des situations envisagées. D'un autre côté, elle devrait s'occuper de la préservation des ressources disponibles (les fonds, le foncier, l'eau, l'énergie et les matières premières) qui devraient être exploitées de manière rationnelle et préservées pour les générations futures.

En effet, les décideurs ont besoin de plus en plus des plans actualisés portants des solutions durables pour faire face aux éventuels changements urbains tout en maintenant les caractéristiques générales des zones urbaines, telles que la couverture, l'interdépendance et la cohérence. Par conséquent, il est nécessaire de faire appel aux experts pour une éventuelle planification urbaine, pour qu'ils fournissent des solutions rationnelles et cela dans les meilleurs délais.

Le présent document propose un système intelligent d'aide à la décision (SIAD), dont l'objectif est de fournir les meilleurs plans urbains associant les projets urbains aux zones appropriées. Ce SIAD s'appuie principalement sur l'approche holonique pour modéliser notre système de planification urbaine, où les agents holoniques de chaque niveau appliquent un nouvel algorithme d'optimisation multi-objectifs par colonies de fourmis (BKPACS) pour résoudre le problème de la planification des projets urbains, considéré comme un problème du sac-à-dos. Afin de produire des plans urbains optimaux globaux, l'algorithme principal appelé H-MACO coordonne entre les différents niveaux de ce système multi-agents holonique (SMAH).

Les expérimentations menées sur les cas d'étude ont montré l'efficacité et la performance des algorithmes proposés pour fournir des plans urbains optimaux, en les comparant à d'autres algorithmes heuristiques et méta-heuristiques notamment les algorithmes génétiques.

### Mots clés :

Système intelligent d'aide à la décision, Système multi-agents holonique, Planification des projets urbains, Développement durable, Problème du sac-à-dos, Algorithme d'optimisation par colonies de fourmis.

## **Abstract:**

Nowadays, urbanization reaches unprecedented spreading, that increases urban problems from day to day, and makes the urban phenomena more dynamic and more complex. Due to decision-makers yearnings and the short time allocated to planners, urban project planning remains an exhausting task; it leads to arbitrary choices to find a good match of projects according to the intended situations. On the other hand, it should take care of the available resources such as funds, land, water, energy, underground, and raw materials, which ought to be rationally exploited, and preserved for future generations.

Actually, decision-makers need more and more updated plans carrying sustainable solutions to face eventual urban changes while maintaining of urban areas features, such as coverage, inter-dependency, and coherency. Therefore, it is important to call in experts for eventual urban planning that provide rational solutions, which often needs to be achieved in less time.

This manuscript proposes an intelligent decision support system (IDSS), which aims to find out the best urban plans that fit urban projects to appropriate areas. The IDSS relies on the holonic approach to model our urban planning system, where the holonic agents of each level apply a new multi-objective ant colony optimization algorithm (BKPACS) to solve the urban project-planning problem, viewed as a knapsack problem (KP). However, to produce global optimal urban plans, the main algorithm called H-MACO coordinates between the different levels of this multi-agents holonic system (MAHS).

The experiments applied on the study cases have shown the efficiency and the performance of the developed algorithms compared to other heuristic and meta-heuristic algorithms such as genetic algorithms, to provide optimal urban plans.

### **Keywords:**

Intelligent Decision Support System, Holonic Multi-agent System, Urban Project Planning, sustainable development, Knapsack Problem, Ant Colony Optimization Algorithm.

## ملخص:

عرف المجال الحضري امتدادا غير مسبوق في الآونة الأخيرة، حاملا معه العديد من المشاكل الحضرية التي ما فتئت تتفاقم من يوم لآخر، مسببة ظاهرة حضرية أكثر ديناميكية وتعقيدا. وبالنظر إلى ما يتوق إليه صناع القرار والوقت المحدود المتاح، يظل تخطيط المشاريع الحضرية مهمة جد مضيئة؛ يلجأ فيها إلى خيارات عشوائية لإيجاد تطابق جيد للمشاريع يتوافق مع الأهداف المرجوة. من ناحية أخرى، ينبغي أن يهتم التخطيط الحضري بالموارد المتوفرة مثل العقار، التمويل، الموارد الطبيعية والطاقوية، والتي يجب استغلالها بطريقة عقلانية حفاظا عليها للأجيال القادمة.

في الواقع يحتاج صناع القرار إلى خطط محينة باستمرار، تتضمن حولا مستدامة لمواجهة كل التغييرات الحضرية المحتملة والتي تراعي الميزات العامة للمناطق الحضرية، مثل التغطية، الترابط والتماسك. لذلك كان من المهم الاستعانة بالخبراء لأجل تخطيط حضري يسمح بتوفير حلول عقلانية في أحسن الأحوال. تقترح هذه الأطروحة نظاما ذكيا لدعم القرار (IDSS)، يهدف إلى تقديم أفضل المخططات الحضرية والتي ترفق المشاريع الحضرية بالمناطق المناسبة لها. إن هذا النظام الذكي يركز أساسا على المقاربة الهولونية *Holonomic approach* لتصميم نظامنا الخاص بالتخطيط الحضري، أين يطبق أعوان النظام الهولوني لكل مستوى خوارزمية جديدة للتحسين المتعدد الأهداف باستعمال مستعمرة النمل (BKPACS) لحل مشكلة تخطيط المشاريع الحضرية التي تعتبر نوعا من أنواع مشاكل حقيبة الظهر (KP). لإنتاج أمثل المخططات الحضرية الشاملة تقوم الخوارزمية المسماة H-MACO بالتنسيق بين مختلف مستويات النظام الهولوني متعدد الأعوان (MAHS).

لقد أظهرت التجارب التي أجريت على عدة حالات النجاعة و الأداء الجيد للخوارزميات المقترحة للحصول على مخططات حضرية مثلى بالمقارنة مع الخوارزميات الإرشادية مثل الخوارزميات الجينية.

## الكلمات الدالة:

نظام دعم إتخاذ القرار الذكي، النظام الهولوني متعدد الأعوان، تخطيط المشاريع الحضرية، التنمية المستدامة، خوارزمية مستعمرة النمل للتحسين، مشكلة حقيبة الظهر.

# Table des matières

Dédicaces .....	i
Remerciements .....	ii
Résumés .....	iii
Table des matières .....	vi
Abréviations .....	x
Liste des tableaux .....	xi
Table des figures.....	xii
<b>INTRODUCTION GÉNÉRALE .....</b>	<b>1</b>
1. Mise en contexte .....	2
2. Motivation .....	3
3. Questions de départ et problématique .....	5
4. Objectifs de la recherche .....	7
5. Esquisse générale de la méthodologie utilisée .....	8
6. Plan de lecture de la thèse .....	9
<b>Chapitre 1 - LA PLANIFICATION DES PROJETS URBAINS .....</b>	<b>11</b>
1.1. Introduction .....	12
1.2. Historique et fondement .....	12
1.2.1. <i>Politique urbaine en Algérie</i> .....	13
1.2.2. <i>Tébessa : patrimoine urbain, particularités et perspectives</i> .....	16
1.3. Pourquoi la planification urbaine est-elle importante ? .....	18
1.3.1. <i>Planifier pour une meilleure occupation des sols</i> .....	19
1.3.2. <i>Planifier pour la satisfaction des citoyens</i> .....	20
1.3.3. <i>Planifier pour changer l'image de la ville</i> .....	20
1.3.3. <i>Planifier pour un développement économique</i> .....	20
1.3.4. <i>Planifier pour protéger l'environnement</i> .....	21
1.4. Organisation du territoire urbain .....	21
1.4.1. <i>Caractéristiques des zones urbaines</i> .....	21
1.4.2. <i>L'échelle spatiale</i> .....	22
1.4.3. <i>L'échelle administrative</i> .....	22
1.4.4. <i>L'échelle sociale</i> .....	22
1.5. Les composantes de base d'un plan urbain .....	22
1.5.1. <i>Occupation de sol</i> .....	23
1.5.2. <i>Logements</i> .....	24
1.5.3. <i>Transport</i> .....	24
1.5.4. <i>Infrastructures de base et services</i> .....	24
1.5.5. <i>Design urbain</i> .....	25
1.5.6. <i>Environnement</i> .....	25
1.6. Le projet urbain et aménagement des villes .....	25
1.6.1. <i>Qu'est-ce qu'un projet urbain ?</i> .....	26



1.6.2. <i>Plan urbain et développement durable</i> .....	27
1.6.3. <i>La démarche de projet urbain</i> .....	28
1.7. Conclusion .....	30
<b>Chapitre 2 - LES SYSTEMES INTELLIGENTS D'AIDE A LA DECISION</b> .....	31
2.1. Introduction .....	32
2.2. Evolution des systèmes d'aide à la décision (SAD) .....	32
2.3. Le passage des SADs vers les SIADs .....	33
2.3.1. <i>Les systèmes à base de connaissances</i> .....	34
2.3.2. <i>Les systèmes à base de réseaux de neurones</i> .....	34
2.3.3. <i>Les systèmes assistés par les agents intelligents</i> .....	35
2.3.4. <i>Les systèmes basés sur la logique floue</i> .....	37
2.3.5. <i>Les systèmes basés sur les ensembles approximatifs (rough sets)</i> .....	38
2.3.6. <i>Les systèmes basés sur l'informatique évolutionnaire</i> .....	40
2.3.7. <i>La fouille des données et la fouille des traitements</i> .....	41
2.3.8. <i>L'apprentissage machine et l'apprentissage profond</i> .....	42
2.4. L'optimisation combinatoire et les SIADs.....	44
2.4.1. <i>Les problèmes classiques de l'optimisation combinatoire</i> .....	44
2.4.2. <i>L'optimisation combinatoire et le problème du sac-à-dos</i> .....	45
2.4.3. <i>Les variantes du problème de sac-à-dos</i> .....	47
2.4.4. <i>La résolution des problèmes d'optimisation combinatoire</i> .....	48
2.4.5. <i>L'optimisation par les colonies de fourmis</i> .....	50
2.5. Synthèse des approches d'optimisation et travaux connexes .....	54
2.6. Conclusion .....	56
<b>Chapitre 3 – UNE CONTRIBUTION BASEE SMAH POUR LA PLANIFICATION DES PROJETS URBAINS</b> .....	57
3.1. Introduction .....	58
3.2. Le concept holon .....	59
3.2.1. <i>Les caractéristiques des holons</i> .....	59
3.2.2. <i>L'architecture holarchique</i> .....	62
3.3. Les systèmes multi-agents (SMA) et l'approche holonique .....	65
3.3.1. <i>Les SMA caractéristiques, avantages et limites</i> .....	66
3.3.2. <i>L'agent holonique</i> .....	68
3.4. Les systèmes multi-agents holoniques (SMAH) et leurs applications : .....	69
3.5. Application de l'approche holonique à la planification des projets urbains .....	71
3.6. Aperçu sur le système urbain .....	71
3.7. Vers une organisation holonique de la planification des projets urbains .....	73
3.8. Modélisation du SMAH de la planification des projets urbains .....	74
3.9. La planification des projets urbains en tant que BKP .....	78
3.10. L'algorithme d'optimisation multi-objectifs par colonies de fourmis du système holonique .....	81
3.10.1. <i>L'information phéromonique</i> .....	82

3.10.2. <i>L'information heuristique</i> .....	83
3.10.3. <i>La sélection probabiliste d'unités de projet</i> .....	83
3.10.4. <i>L'algorithme BKPACS</i> .....	84
3.10.5. <i>L'algorithme H-MACO</i> .....	84
3.11. Description et fonctionnement du SIAD-PPU .....	86
3.12. Conclusion .....	88
<b>Chapitre 4 – EXPERIMENTATION ET EVALUATION</b> .....	<b>90</b>
4.1. Introduction .....	91
4.2. Le prototype SIAD-PPU .....	91
4.2.1. <i>Administration des données</i> .....	92
4.2.2. <i>Initialisation des paramètres</i> .....	92
4.2.3. <i>La formulation des requêtes</i> .....	93
4.2.4. <i>L'optimisation des plans des projets urbains</i> .....	93
4.3. Apports et limites du SIAD-PPU .....	95
4.3.1. <i>Expérimentations et discussion des résultats</i> .....	95
4.3.2. <i>Limites du SIAD-PPU</i> .....	101
4.4. Conclusion.....	101
<b>CONCLUSION GENERALE ET PERSPECTIVES</b> .....	<b>102</b>
1. Bilan de la thèse .....	103
2. Perspectives de la recherche .....	104
<b>Bibliographie</b> .....	<b>106</b>
<b>Annexes</b> .....	<b>118</b>
A. Liste des publications .....	119
B. Sources .....	120

## Abréviations

ACS	ant colony system ( <i>Eng.</i> )
AEP	adduction d'eau potable, ou alimentation en eau potable
AG	algorithmes génétiques
APC	assemblée Populaire Communale
AS	ant system ( <i>Eng.</i> )
BA	bat algorithm ( <i>Eng.</i> )
BI	business intelligence ( <i>Eng.</i> )
BKP	bounded knapsack problem ( <i>Eng.</i> )
BKPACS	ant colony system for bounded knapsack problem ( <i>Eng.</i> )
COP	combinatorial optimization problem ( <i>Eng.</i> )
CSO	competitive swarm optimizer ( <i>Eng.</i> )
d-KP	multidimensional knapsack problem
EHO	elephant herd optimization ( <i>Eng.</i> )
GRASP	greedy randomized adaptive search procedures ( <i>Eng.</i> )
H-MACO	holonic multi-objective ant colony optimization ( <i>Eng.</i> )
IA	intelligence artificielle
IDSS	intelligent decision support system ( <i>Eng.</i> )
IFM	Information Feedback Models ( <i>Eng.</i> )
JSON	Javascript Object Notation ( <i>Eng.</i> )
QKP	quadratic knapsack problem
KDD	knowledge discovery in database ( <i>Eng.</i> )
KP	knapsack problem ( <i>Eng.</i> )
LF	logique floue
MCKP	multiple choice knapsack problem
MDO	multidisciplinary design optimization ( <i>Eng.</i> )
MKP	multiple knapsack problem ( <i>Eng.</i> )
ML	machine learning ( <i>Eng.</i> )
MOEA/D	multi-objective evolutionary algorithm based on decomposition ( <i>Eng.</i> )
MOKP	multi-objective knapsack problem MOKP ( <i>Eng.</i> )
MOO	multi-objective optimization MOO ( <i>Eng.</i> )
MOP	multi-objective problem ( <i>Eng.</i> )
NP-difficile	non polynomial-difficile
OCF	optimisation par les colonies de fourmis
OLAP	online analytical processing ( <i>Eng.</i> )
PAW	plans d'aménagement de wilaya
PCD	plan communal de développement
PDAU	plan directeur d'aménagement et d'urbanisme
PE	programmation évolutionnaire
PG	programmation génétique
PMU	plan de modernisation urbaine
POS	plans d'occupation des sols
PPU	planification des projets urbains
PSD	plan sectoriel de développement
PU	projet urbain

PUD	plan d'urbanisme directeur
RBC	raisonnement à base des cas
RNA	réseaux de neurones artificiels
SAD	système d'aide à la décision
SBC	système à base de connaissances
SF	système flou
SIAD	système intelligent d'aide à la décision
SIAD-PPU	système intelligent d'aide à la décision pour la planification des projets urbains
SIG	système d'information géographique
SMA	système multi-agents
SMAH	système multi-agents holoniques
SNAT	schéma national d'aménagement du territoire
SOHO	self-regulating open hierarchic order ( <i>Eng.</i> )
SRAT	schéma régional d'aménagement du territoire
TSP	traveling salesman problem ( <i>Eng.</i> )
OCF	optimisation par les colonies de fourmis
XML	eXtensible Markup Language ( <i>Eng.</i> )

## Liste des tableaux

**Tableau 3.1** Holon vs Agent

**Tableau 4.1** Les données des projets d'une sous-zone urbaine

**Tableau 4.2** Catalogue des projets urbains

**Tableau 4.3** Comparaison des plans pseudo-optimaux des projets urbains générés par l'algorithme BKPACS par rapport à l'algorithme génétique

**Tableau 4.4** Evaluation du temps moyen d'optimisation global des plans des projets urbains par l'algorithme H-MACO

**Tableau 4.5** Résultat récapitulatif des expériences sur l'algorithme H-MACO

## ✚ Table des figures

<b>Figure 0.1</b>	Affectation des projets à une zone urbaine
<b>Figure 1.1</b>	Situation géographique et Administrative de la wilaya de Tébessa
<b>Figure 1.2</b>	Principaux champs d'intervention de la planification urbaine
<b>Figure 1.3</b>	Composantes de base d'un territoire urbain
<b>Figure 1.4</b>	Démarche et étapes du projet urbain
<b>Figure 2.1</b>	Architecture d'un système à base de connaissances
<b>Figure 2.2</b>	Perceptron biologique VS perceptron artificiel
<b>Figure 2.3</b>	Les réseaux de neurones artificiels
<b>Figure 2.4</b>	Les architectures hiérarchique, holarchique, et hétérarchique des SMA
<b>Figure 2.5</b>	Structure générale d'un système d'inférence floue
<b>Figure 2.6</b>	Ensembles approximatifs ( <i>Rough set</i> )
<b>Figure 2.7</b>	Diagramme général de l'algorithme génétique
<b>Figure 2.8</b>	Processus de la découverte de connaissances dans une base de données
<b>Figure 2.9</b>	Les types d'apprentissage machine
<b>Figure 2.10</b>	Comportement des fourmis à la recherche de la nourriture
<b>Figure 3.1</b>	L'holon dans une structure
<b>Figure 3.2</b>	Organisation holarchique comme étant une structure multicouche
<b>Figure 3.3</b>	Architecture d'une holarchie multicouches
<b>Figure 3.4</b>	Modèle d'holarchie descendant
<b>Figure 3.5</b>	Composition des agents holoniques dans un SMAH
<b>Figure 3.6</b>	Partitionnement du territoire en zones urbaines
<b>Figure 3.7</b>	Représentation du territoire urbain en multicouches
<b>Figure 3.8</b>	Problème d'affectation de projets urbains par secteur d'activité aux zones urbaines
<b>Figure 3.9</b>	Vue globale du réseau SMAH de la planification des projets urbains
<b>Figure 3.10</b>	L'architecture holonique proposée pour le système de planification des projets urbains
<b>Figure 3.11</b>	Système holonique de la planification des projets urbains
<b>Figure 3.12</b>	La structure de la phéromone pour la planification des projets urbains
<b>Figure 3.13</b>	Système intelligent d'aide à la décision basée sur l'approche holonique pour la planification des projets urbains
<b>Figure 4.1</b>	Interface principale du prototype SIAD-PPU
<b>Figure 4.2</b>	Initialisation des paramètres et déroulement d'un exemple d'optimisation par l'algorithme BKPACS
<b>Figure 4.3</b>	Simulation d'optimisation des projets urbains par l'algorithme BKPACS
<b>Figure 4.4</b>	L'information phéromonique lors d'un processus d'optimisation par l'algorithme BKPACS
<b>Figure 4.5</b>	Optimisation des plans des projets urbains par l'algorithme génétique

- Figure 4.6** Une variété de plans des projets urbains générés par l'algorithme BKPACS comparés à celui généré par l'algorithme *greedy simple*
- Figure 4.7** Convergence des solutions par l'algorithme BKPACS
- Figure 4.8** Variation du temps moyen d'optimisation global par l'algorithme H-MACO

# Introduction générale





# INTRODUCTION GENERALE

## 1. Mise en contexte

La ville, c'est l'endroit où je vis, je travaille, je me promène, je rencontre des amis, et je fais tant d'activités pour profiter des nombreux services offerts tous les jours. Et comme définit par *Hardt et Negri* [Lecoq (2017)], « la ville est mon contexte de vie et que, de ce fait, je me définis comme membre d'une société urbaine de plus en plus dense et nombreuse, la ville est alors, pour moi, mon contexte de lutte ». Les logements, les réseaux de transport, les infrastructures de base et services publics, les centres d'activités et de loisir constituent les composantes principales de la ville et dessinent le quotidien du citoyen.

Afin de respecter le cadre de vie des habitants, la mixité urbaine [Legrand and Meunier-Chabert (2003)], l'équilibre et la coexistence des populations, et la préservation de l'environnement, il est nécessaire de planifier le territoire urbain pour coordonner le développement et la création des villes, et cela par la mise en place des mécanismes de contrôle des différentes activités qui influent sur la composition urbaine et sur le développement de la ville.

Cette pratique urbanistique s'articule sur l'analyse du territoire et le diagnostic de l'état présent de la communauté, qui conditionnent la prise de décision. Il faudrait bien comprendre le fonctionnement de la communauté urbaine et savoir à quel point elle tend vers le changement pour dresser de meilleurs plans qui assurent une bonne répartition des composantes urbaines.

Grâce au plan de la ville (plan urbain) [Gaudin (1985)], on peut avoir un aperçu global du fonctionnement de la ville en termes d'infrastructures, routes, style d'immeubles, parcs etc., et une vision future montrant en quoi ça peut ressembler cette ville dans l'avenir en tenant compte de l'emploi, de l'activité économique, du nombre d'habitants, et des moyens de transport.

En fait, la planification urbaine vise à fournir des idées, des réponses, et des solutions aux problèmes des cités, des villes, et des métropoles. Elle permet aux pouvoirs publics et privés grâce aux instruments d'urbanisme de prévoir, d'orienter, d'organiser et de maîtriser le développement du territoire urbain.

En réaction à l'urbanisme fonctionnaliste des années 60 [Dureau (1993)], et opposant les approches sectorielles [Gwiazdzinski (2015)] de la planification urbaine, le projet urbain (PU) veille d'une part au maintien des acquis en matière d'urbanisme sur le niveau local, et d'autre part à la mise en place de futurs projets de développement et d'aménagement du territoire. C'est un facteur primordial pour le succès de tout développement durable en impliquant l'ensemble des acteurs concernés par le processus de planification urbaine tels que les décideurs, les bailleurs de fond, les gestionnaires, les concepteurs, les promoteurs, les investisseurs, les intellectuels, les chercheurs, et les citoyens pour réunir les idées vers une vision globale de l'avenir de la ville.

A travers la concertation et la participation avec l'ensemble de ces acteurs, les projets consensuels nécessitent d'identifier les potentialités, les problèmes de la ville, et les enjeux majeurs en vue d'atteindre les objectifs d'une planification stratégique. En fait, le projet urbain [Toussaint and Zimmermann (1998)] constitue une projection des activités et aménagements urbains à entreprendre aussi bien dans l'espace que dans le temps. Certes, le plan urbain correspondant doit assurer la gestion et la préservation des ressources naturelles, foncières et financières.

Le domaine urbain évolue constamment vu que les instruments classiques d'urbanisme ont montré leurs échecs pour maîtriser la complexité du processus urbanistique due au cumul des problèmes relevant des contextes géographiques, historiques, politiques, et socio-économiques [McLoughlin (1969)]. Par conséquent l'adoption de nouvelles méthodes, démarches et outils de planification et d'aide à la décision demeure un enjeu majeur.

Enfin, l'avènement des nouvelles technologies ont permis à la communauté urbaine d'intégrer des systèmes informatiques dans la gestion du territoire urbain et ses ressources [Laurini (2003)]. Néanmoins, la planification des projets urbains reste une tâche fastidieuse, car elle s'appuie principalement sur des choix combinatoires pour trouver de bonnes adéquations des projets aux zones appropriées et cela dans les délais impartis.

Par ailleurs, l'évolution de l'intelligence artificielle conjointement aux techniques d'aide à la décision s'impose dans tous les domaines. Elle se traduit par le développement et la mise en place des systèmes intelligents d'aide à la décision (SIAD) dont les fonctionnalités varient pour donner des résultats compétitifs.

Dans cette thèse, le SIAD proposé s'articule principalement sur l'approche holonique offrant un modèle hiérarchique adéquat au système urbain, et permettant de modéliser ses composantes sous forme d'holons. Autour des différents critères et contraintes, ce système vise à fournir des plans optimaux qui affectent rationnellement les projets urbains aux zones appropriées dans le temps.

## **2. Motivation**

Le système urbain évolue tellement vite que les outils de la planification urbaine se montrent incapables de contourner les problèmes urbains et satisfaire les besoins urbanistiques de la communauté. Surtout avec la croissance démographique, la gestion des composantes urbaines et l'exploitation des ressources deviennent une tâche complexe.

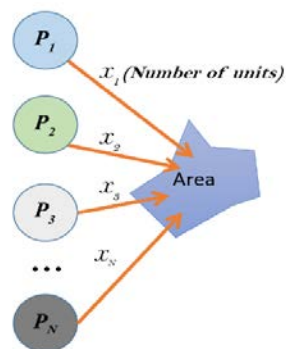
D'autre part, les problèmes urbains mettent les collectivités locales dans l'impasse, donc elles s'investissent à trouver des solutions immédiates aux situations critiques. Néanmoins, intervenir hâtivement dans de telles situations pourrait conduire à des solutions arbitraires, qui engendrent des conflits et des conséquences négatives tant sur le niveau local que sur le niveau global du territoire urbain.

Le processus de la planification urbaine contribue à la gestion du milieu urbain, en offrant des solutions sur les plans organisationnel et opérationnel vers une vision globale de l'avenir de la ville [Bacqué and Gauthier (2011)]. Il veille d'une part à fournir des plans urbains adéquats et d'autre part au suivi et à la mise en œuvre de ces plans.

Quels que soient le domaine d'application, la décision d'individus ou de groupe, et l'objectif attendu, les techniques classiques d'aide à la décision s'appuient principalement sur l'analyse, l'évaluation et la comparaison des critères et/ou attributs propres aux différentes alternatives [Simon (1977)]. Cependant ces techniques restent limitées particulièrement envers les problèmes complexes et les problèmes de grande dimension notamment les problèmes d'optimisation combinatoire.

Elaborer un plan des projets urbains suivant la méthodologie proposée dans cette thèse, revient à résoudre un problème d'optimisation combinatoire auquel on peut associer un problème de décision, dont le but est de chercher s'il existe une solution optimale voir même pseudo-optimale pour laquelle la fonction multi-objectifs – une fonction qui réunit plusieurs objectifs (par exemple : le coût, le temps, la qualité, l'impact, etc.) à optimiser – soit vérifiée. En effet, la complexité d'un problème d'optimisation dépend de celle du problème de décision qui lui est associé. D'autant plus, si le problème de décision est *NP-complet*, alors son optimisation devient *NP-difficile* [Ausiello et al. (1995), Benyamina (2013)].

En général, dans un processus de planification des projets urbains, l'affectation de certains sous-ensembles de projets peut être analysée par une approche de type sac-à-dos<sup>1</sup>, qui consiste à répartir de manière rationnelle  $N$  projets de développement à une zone urbaine, de telle sorte que cette affectation satisfasse l'objectif et respecte les contraintes. Pour résoudre ce problème, il s'agit d'énumérer exhaustivement toutes les combinaisons possibles des solutions dans l'espace de recherche, sans écarter aucune solution supposée être meilleure que la solution optimale trouvée.



**Figure 0.1** – Affectation des projets à une zone urbaine  
(Source : [Khelifa and Laouar (2018)])

---

<sup>1</sup> Le problème du sac à dos modélise une situation analogue au remplissage d'un sac à dos, qui ne peut supporter plus d'un certain poids, avec un ensemble d'objets ayant chacun un poids et une valeur. Les objets mis dans le sac à dos doivent maximiser la valeur totale, sans dépasser le poids maximum du sac. Ce problème fait partie des 21 problèmes *NP-complets* identifiés par *Richard Karp* en 1972 [Karp (1972)].

Ça pourrait être souvent le cas où tous les projets ne sont pas différents les uns des autres, c'est pour cela un nombre d'unités identiques de chaque projet est donné [Kellerer et al. (2004)]. S'il est possible de sélectionner un sous ensemble d'unités de chaque projet, on pourrait soit représenter chaque unité par une variable binaire, ou plus efficacement représenter l'ensemble entière des unités identiques d'un projet  $P_i$  par un entier positif  $x_i$  indiquant le nombre d'unités sélectionnées de ce projet (Figure 0.1).

Ici, l'affectation des projets urbains est plutôt un problème du *sac-à-dos borné*, qui se ramène à un problème d'optimisation complexe *NP*-difficile pour lequel les *méthodes exactes* et les *méthodes d'approximation* s'avèrent très vite impraticables surtout pour planifier un territoire urbain de grande dimension [Ausiello et al. (1995), Benyamina (2013)]. Le choix d'un projet dépend pratiquement souvent de l'optimisation de plusieurs objectifs. Ce qui mène à un problème du sac-à-dos multi-objectif (*multi-objective knapsack problem* MOKP).

Dans la démarche du projet urbain [Avitabile (2005)], la planification ne se limite point à une seule zone urbaine, mais elle pourrait s'étendre sur tout le territoire pour réaliser les objectifs globaux. Cette situation conduit à la variante du sac-à-dos multiple (*multiple knapsack problem* MKP) qui consiste à répartir un ensemble d'unités de projets sur plusieurs zones urbaines différentes, sachant que la valeur d'un projet dépend de la zone à laquelle il est affecté.

Certainement, les méthodes classiques d'optimisation servent à décomposer les problèmes complexes en sous problèmes et à définir les démarches de résolution, par lesquelles ils ont prouvé leur efficacité, mais face aux difficultés dues à la complexité croissante des problèmes, à la distribution des données, à la nature dynamique et incertaine du domaine, et à la diversité des besoins, il est devenu nécessaire de chercher de nouvelles solutions en explorant de nouveaux paradigmes que ceux habituellement utilisés.

Les décideurs sont alors motivés à réfléchir à changer les méthodes traditionnelles utilisées dans la planification des projets urbains et faire appel aux nouvelles technologies, qui ont connu un grand succès dans plusieurs domaines, grâce aux techniques intelligentes fiables et efficaces.

### **3. Questions de départ et problématique**

Dans le modèle traditionnel de l'urbanisme, le développement commence par une conception abstraite des plans urbains en suivant une approche sectorielle de la planification urbaine [Gallez and Maksim (2007)]. Cette approche décrit ce que le système doit faire, et quelles sont ses fonctionnalités. Autrement dit, les plans urbains étaient élaborés selon la vision commune des concepteurs et urbanistes, sans impliquer le citoyen et les autres parties prenantes. Malheureusement, de nombreux projets urbains ont été abandonnés, pourtant ils ont été bien construits sur le plan architectural, car ils ne correspondent pas aux besoins des utilisateurs.

Donc, la planification hiérarchique ne garantit pas que le système proposé réponde parfaitement aux attentes de la communauté de son usage. En fait, l'échec de la mise en œuvre des plans est généralement dû à une mauvaise prise en charge des besoins auxquels le système urbain tente de répondre. Par ailleurs, la dominance du pouvoir centralisé ne permet pas assez de liberté d'actions aux collectivités locales [Epstein (2013)].

L'avènement du projet urbain constitue une révolution dans l'urbanisme de la planification et une alternative opposant les approches traditionnelles. Le projet urbain offre plus d'ouverture et de flexibilité, de façon à ce que les acteurs potentiels participent pleinement dans le processus de développement à travers le pouvoir décisionnel acquis [Pinson (2009)]. La démarche du projet urbain tente de rapprocher les idées vers des compromis, qui réunissent la communauté urbaine, dont les opinions et les intérêts sont différents voire même conflictuels. En employant cette démarche, la volonté de changer la ville s'accroît en effectuant un minimum d'actions dans le temps par rapport aux objectifs auxquels tous les acteurs concernés doivent coopérer simultanément.

Bien que l'approche du projet urbain soit dotée d'outils de planification performants à l'échelle locale et à l'échelle globale, il faudrait tout de même soutenir cette approche par l'emploi d'autres outils et moyens technologiques, qui ont prouvé leur efficacité dans divers domaines notamment l'industrie, la médecine, et le commerce, et comment bénéficier de l'avancée de l'intelligence artificielle pour recommander aux décideurs de meilleurs plans de projets urbains durables équilibrés et diversifiés ?

Après une lecture consciente des principes de base de la planification urbaine, suivie d'une étude approfondie sur les systèmes intelligents d'aide à la décision ainsi que les approches organisationnelles appliquées aux systèmes complexes distribués, il convient maintenant de se poser la question suivante : Comment pouvons-nous utiliser tous ces éléments et en tirer profit pour mettre en place un système intelligent d'aide à la décision pour la planification de projets urbains ?

Le système urbain possède tellement de spécificités qu'il se range dans la catégorie des domaines dynamiques, complexes, et à grande échelle. Donc, appliquer les outils d'aide à la décision représente une tâche difficile, cela évoque les questions suivantes : est-il nécessaire de décomposer ce système en simples entités urbaines ? et quel type d'organisation faudra-t-il choisir ?

Certainement, les structures hiérarchiques permettent une vision globale et claire du système étudié en montrant les détails de ses composantes. Parmi d'autres la structure holonique offrant plus d'ouverture et de flexibilité via le concept holon conçu par Koestler [Koestler (1967)], qui considérerait la relation tout/partie entre les entités, et définit l'holon comme étant un tout faisant partie d'un tout plus vaste, et qui contient en même temps des sous parties par lesquelles il est composé. En comparant la hiérarchie urbaine avec l'holarchie de Koestler, cette organisation holonique est-elle capable de modéliser le système urbain en tenant compte de sa dimension et de sa complexité ?

D'autant plus, la modélisation du système nécessite de décrire clairement le fonctionnement de tous ses composants, en mettant en évidence les interactions entre ses entités et les traitements à effectuer par chacune. Dans cette phase, on doit définir les processus du système, les techniques à employer, et les modifications possibles à apporter aux algorithmes relatifs.

Toutes ces questions définissent notre problématique de recherche, et ouvrent un débat qui met en lien différents concepts de la planification urbaine ainsi que les outils d'aide à la décision et de l'intelligence artificielle.

#### **4. Objectifs de la recherche**

Tant pour les grandes villes, que les petites villes, les problèmes varient selon les contextes géographiques, historiques, politiques, économiques, et sociaux de chacune. Par conséquent l'adoption de nouvelles méthodes, démarches et outils de planification et d'aide à la décision constitue un défi majeur.

Face aux contraintes liées à la spécificité du domaine urbain, il est important de choisir une organisation capable de modéliser ce système, et prendre en compte de sa dimension et de sa complexité, l'organisation holonique s'avère la plus appropriée. Etant fondée sur l'holon qui est une extension du paradigme agent intelligent, cette organisation nous a permis de développer, d'intégrer un algorithme méta-heuristique et d'en bénéficier dans l'optimisation des plans des projets urbains.

Notre travail s'inscrit dans le domaine d'aide à la décision, et plus précisément au développement d'un système intelligent d'aide à la décision (SIAD) pour la planification des projets urbains, et ce en employant les approches, les méthodes, et les outils nécessaires à l'organisation, à la modélisation et au développement de ce système. Ce SIAD devrait combiner les connaissances du domaine, avec les données relatives aux citoyens, aux infrastructures et aux services afin d'assister la communauté urbaine (les décideurs, les experts urbains, et les planificateurs) à élaborer des plans urbains optimaux, qui pourraient satisfaire les besoins des citoyens, et répondent aux exigences du niveau local et aux directives du niveau global.

Ces projets appartiennent aux divers programmes du développement centralisés et/ou décentralisés, publics ou privés, annuels voir même quinquennaux, sont le fruit d'un long processus d'étude, de planification et de concertation. Ils couvrent plusieurs classes dans différents secteurs d'activités. En effet, notre SIAD s'intéresse essentiellement aux projets d'intérêt général à court et à moyen terme sur le plan local. Malheureusement, certains projets se font sur le long terme comme les élargissements de rues, les lignes de tram ou de métro, qui doivent être traités séparément, car ils nécessitent l'étude de variantes de localisation.

La mise en place de ce SIAD part de l'hypothèse fondée sur le succès de l'organisation holonique dans plusieurs domaines notamment l'industrie manufacturière. De même l'évolution impressionnante de l'intelligence artificielle nous a motivé à tirer profit des techniques d'optimisation et particulièrement des algorithmes méta-heuristiques afin d'optimiser les plans des projets urbains.

Pour valider notre hypothèse, nous avons souligné les objectifs suivants :

- Etudier le domaine de la planification urbaine, afin de mieux connaître ses composantes, ses caractéristiques, son organisation, et comment on peut intervenir avec des moyens technologiques à planifier les projets urbains.
- Exposer l'état de l'art des SAD et des SIAD ainsi que les approches d'optimisation combinatoire.
- Explorer les concepts relatifs à l'approche holonique afin de concevoir un système multi-agents holoniques (SMAH) dédié à la planification des projets urbains.
- Proposer une architecture basée SMAH pour notre système de planification des projets urbains.
- Développer des algorithmes en faveur des agents holoniques formant ce système.
- Développer un prototype pour le test et l'évaluation de l'efficacité des algorithmes à travers des cas d'étude.

## **5. Esquisse générale de la méthodologie utilisée**

La planification des projets urbains faisant objet de notre étude, ressemble en quelque sorte au problème du sac-à-dos (*knapsack problem* KP), qui est largement utilisé dans divers domaines, et particulièrement dans l'allocation des ressources. Dans notre cas, chaque projet comprend un nombre *limité* d'unités identiques caractérisées par leurs préférences et leurs coûts, nous avons donc opté pour la variante sac-à-dos borné (*bounded knapsack* BKP), permettant d'optimiser les plans urbains par une sélection rationnelle d'unités de projets, afin de satisfaire les besoins de la communauté urbaine, et de respecter en même temps les contraintes limitant l'exploitation des ressources.

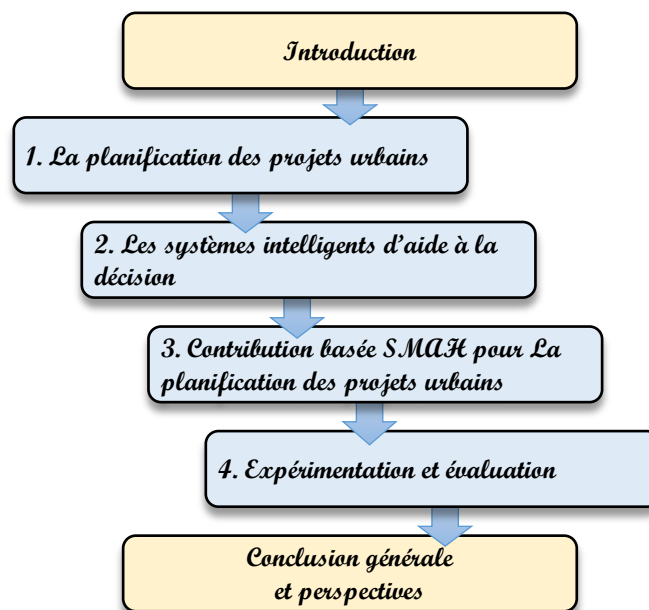
Etant donné, le développement d'un territoire urbain s'appuie sur la décomposition administrative, ainsi pour promouvoir une zone urbaine à travers la démarche du projet urbain, nécessite de prendre en considération toutes les zones urbaines adjacentes, qui doivent collaborer ensemble afin d'assurer l'intégrité du système urbain.

En principe, l'organisation en zones urbaines semble être facile à gérer et à planifier en matière d'infrastructures et d'équipements, mais elle présente des insuffisances pour réaliser les objectifs stratégiques dans des systèmes urbains complexes et/ou à grande échelle. Il faudra alors recourir à un modèle d'organisation plus flexible capable à la fois de conserver l'autonomie et de garantir la dépendance de la zone urbaine, et par conséquent choisir librement sa stratégie dans son environnement.

Ces caractéristiques favorisent le choix d'une organisation holonique, ce qui nous a conduit à proposer un système multi-agents holonique (SMAH) destiné à planifier les projets urbains, dont les agents holoniques appliquent un nouvel algorithme méta-heuristique d'optimisation multi-objectifs par colonies de fourmis appelé BKPACS. Etant donné que la construction des plans des projets urbains optimaux au niveau local constitue un problème de sac-à-dos borné, nous avons développé dans le cadre de cette thèse l'algorithme BKPACS spécialement pour résoudre ce problème. Tandis que la coordination entre les différents niveaux de l'holarchie et l'optimisation de l'ensemble des plans urbains sur le niveau global sont assurés par l'algorithme H-MACO.

## 6. Plan de lecture de la thèse

Notre travail est structuré en quatre chapitres comme décrit dans la figure suivante :



Ainsi, la thèse est organisée comme suit :

☞ L'introduction générale présente le contexte de travail, les motivations de la recherche, la problématique, une esquisse générale de la méthodologie utilisée, et les objectifs souhaités ;

☞ Le premier chapitre introduit les concepts et principes de base de la planification urbaine. Il décrit la politique urbaine dans le contexte national et local, puis il détaille l'utilité de la planification urbaine et ses composantes formant l'ossature de la ville, ensuite ce chapitre se focalise sur l'importance du projet urbain dans le développement durable, ainsi que la démarche du projet urbain qui constitue le centre d'intérêt de notre étude ;

☞ Le second chapitre est consacré aux systèmes intelligents d'aide à la décision (SIAD) y compris les méthodes d'optimisation telles que les approches heuristiques et méta-heuristiques. En premier, nous présentons l'évolution des systèmes d'aide à la décision (SAD), le passage vers les SIAD, et les principales techniques employées dans les SIAD, puis nous évoquons la relation entre l'optimisation et les SIAD à travers laquelle on




montre son utilité dans la résolution des problèmes décisionnels, combinatoires et complexes. Nous y exposons brièvement les problèmes classiques d'optimisation combinatoire, en particulier le problème de sac-à-dos – qui servira par la suite à modéliser la planification des projets urbains – ainsi que les approches d'optimisation (exactes, d'approximation et heuristiques) notamment l'optimisation par les colonies de fourmis. Ce chapitre est également enrichi par un état de l'art couvrant les travaux connexes ;

☞ Le troisième chapitre couvre nos contributions concernant la planification des projets urbains en se basant essentiellement sur l'approche holonique et les algorithmes d'optimisation méta-heuristiques et leur application sur les projets urbains considérés comme étant un problème de sac-à-dos borné, qui pourrait lors d'une planification globale du territoire s'étendre vers un problème de sac-à-dos multiple.

Avant d'entamer la modélisation de notre système intelligent d'aide à la décision de la planification des projets urbains, il est indispensable de définir le concept d'holon, ses caractéristiques, son organisation, sa structure, ainsi que l'architecture holarchique. Ce qui nous a aidé à concevoir un système multi-agents holonique (SMAH) de la planification des projets urbains. Ce chapitre propose d'avantage un support technique d'algorithmes (BKPACS, H-MACO) développés spécialement en faveur des agents holoniques et une description détaillée du fonctionnement de ce système.

☞ Le quatrième chapitre est dédié à l'analyse et à l'évaluation du système intelligent d'aide à la décision de la planification des projets urbains. En premier lieu, nous présentons le prototype développé pour tester les algorithmes BKACS et HMACO, en mettant en évidence l'utilisabilité de l'approche holonique conjointement avec les techniques d'optimisation méta-heuristiques par colonies de fourmis, pour résoudre des problèmes décisionnels complexes et à grande échelle. Puis nous passons aux expérimentations sur des cas d'étude élaboré suivant les caractéristiques d'un territoire urbain de la wilaya de Tébessa. L'efficacité de nos algorithmes pour fournir des plans optimaux urbains fera l'objet d'une comparaison à d'autres algorithmes heuristiques et méta-heuristiques notamment les algorithmes génétiques.

☞ A la fin de cette thèse, une conclusion générale résumant l'ensemble de nos travaux et décrivant les perspectives envisagées.

 *Chapitre 1*

# La planification des projets urbains



# Chapitre - 1

## LA PLANIFICATION DES PROJETS URBAINS

### 1.1. Introduction

La planification urbaine qui s'intéresse principalement à exposer ce qui est bon et ce qui est mauvais pour la société, et comment elle pourrait maintenir son existence afin de se perfectionner dans l'avenir. C'est une méthode qui désormais permet aux pouvoirs publics grâce aux instruments d'urbanisme de prévoir, d'orienter, d'organiser et de maîtriser le développement du territoire urbain.

Cette pratique urbanistique s'articule sur l'analyse du territoire et le diagnostic de l'état présent de la communauté qui conditionnent la prise de décisions publiques et privées afin d'atteindre l'objectif urbain souhaité par le projet qui figure dans le plan urbain.

Les moyens nécessaires pour élaborer un plan urbain dépend du type du plan établi et de la dimension de la ville, certes une petite cité pourrait admettre un bon plan urbain en une courte durée pour un montant relativement modeste, par contre un plan urbain pour une grande cité pourra prendre des années et nécessite l'intervention d'un grand nombre de personnes ainsi qu'un budget très important. Pour dresser de meilleurs plans, il faudrait bien comprendre le fonctionnement de la communauté urbaine et savoir à quel point elle tend vers le changement.

Ce chapitre introduit les principes de base de la planification urbaine et la manière dont celle-ci fonctionne. Il s'agit aussi de définir les raisons pour lesquelles les communautés veillent à la mise en œuvre des plans urbains, les acteurs agissant dans le milieu urbain, les caractéristiques d'un découpage urbain, le projet urbain comme étant une démarche planificatrice répandue et galonnée, ainsi que les défis auxquels sont confrontées les villes et les communautés d'aujourd'hui.

### 1.2. Historique et fondement

C'est dans la deuxième moitié du 19<sup>ème</sup> siècle que le concept de la planification urbaine a vu le jour, à l'issue de la révolution industrielle en Europe entraînant des problèmes qui ont créé des villes anarchiques et polluées. Ce qui donné naissance à de nouveaux concepts tels que « *city planning, community planning, town planning, regional planning* » [Yin (2012)] permettant de s'opposer à la ville industrielle et de mettre en place des mesures d'hygiène et de morales. Au début du 20<sup>ème</sup> siècle, ces concepts ont été appliqués notamment en grande Bretagne « Letchworth, Welwyn » puis en France et aux États-Unis d'Amérique « Chicago, new York ».

En 1930, de nouvelles politiques d'aménagement de territoire apparaissent et se développent pour reconstruire les villes de l'après-guerre mondiale (1939-1945) [Taylor (1998)]. Cette opération d'urgence a été marquée par un manque d'organisation et de vision à long terme [Lakhdar Hamina and Abbas (2015)].

Dès les années 1970 les urbanistes et architectes ont introduit la notion de projet par opposition à la notion de plan relevant des pratiques de l'urbanisme fonctionnaliste et technocratique [Pinson (2006)]. Cette transition se base sur l'éradication des anciens quartiers, l'industrialisation de la production du logement, la séparation des fonctions ainsi que revoir les rapports décideurs-concepteurs d'une part et les citoyens d'une autre part.

Au cours des années 1970-80 des initiatives se développent pour mettre en place des méthodes alternatives de conception avec l'association des habitants. L'apparition du projet urbain - comme une nouvelle approche succédant la planification urbaine - vise à mieux gérer la ville, et à répondre aux attentes des citoyens sur tous les plans social, économique, et environnemental.

Faire du projet urbain, c'est préférer l'urbanisation intensive plutôt que l'urbanisation extensive, c'est-à-dire s'inspirer de l'existant pour construire la ville, reproduire ses typologies et trames et renouveler son tissu urbain de manière à maintenir son identité. L'association des habitants conduit à un consensus, permettant de mieux concevoir et bien décider au profit des usagers.

Après le sommet de la terre à RIO en 1992 et la campagne des villes européennes de Aalborg en 1994, de nouveaux concepts notamment ville durable et gouvernance urbaine [Pinson (2006), Lakhdar Hamina and Abbas (2015)] s'ajoutent à d'autres : projet d'agglomération, projet de ville [Pinson (2006)], projets urbains, projet de quartier [Pinson (2005), Courcier (2005)].

### ***1.2.1. Politique urbaine en Algérie***

L'Algérie a connu plusieurs changements urbains dus aux époques de civilisations successives notamment la civilisation romaine, byzantine, musulmane et ottomane ainsi que la colonisation française qui a appliqué une réglementation administrative typiquement française en Algérie.

Durant la période 1830-1919, la politique urbaine était consacrée à l'application d'un *plan d'alignement et des réserves* dont l'objectif essentiel était de fixer les alignements des rues nouvelles, des espaces publics, des réserves foncières ainsi que la création des différents lotissements [Lakhdar Hamina and Abbas (2015)].

Après la première guerre mondiale, la France a mis en place la loi dite *Cornudet* (1919-1924) concernant même ses colonies, conçue particulièrement pour le développement des agglomérations dans une perspective générale et cohérente de la ville et son extension à travers un *plan d'aménagement, d'extension et d'embellissement* suite au décret du 05/01/1922.

En 1948, des nouvelles mesures pour la planification urbaine en Algérie par un recensement des problèmes d'urbanisme basé sur la méthode *des ratios et la grille des équipements* en donnant plus d'importance aux différents réseaux (réseaux

d'assainissement, réseaux de transports, réseaux d'eau potable) afin d'améliorer la vie sociale et économique et de rejoindre le niveau de modernisation en France.

Vers la fin de la colonisation, plusieurs instruments d'urbanisme ont été mis en place dont l'importance a influencé les instruments de l'après indépendance. Issus de l'application des lois : 87-03 du 27 Janvier 1987 relative à l'aménagement du territoire, 90-29 du 01 décembre 1990 relative à l'aménagement et l'urbanisme ainsi que la loi 01-20 du 12 décembre 2001 relative à l'aménagement du territoire et au développement durable, ces instruments se traduisent en documents de planification et d'urbanisme énumérés comme suit :

- Le schéma national d'aménagement du territoire (SNAT) : Introduit par la loi 87-03 modifiée et complétée par la loi 01-20, c'est un plan global d'exploitation et d'aménagement dans une perspective générale qui couvre le territoire national à long terme, évalué et actualisé périodiquement. Par une concertation des acteurs concernés, le SNAT vise une répartition équitable des richesses, des activités et du peuplement ainsi qu'une localisation des grandes infrastructures. Ainsi le SNAT prépare l'élaboration et contrôle l'exécution des schémas régionaux d'aménagement du territoire (SRAT) ;
- Le schéma régional d'aménagement du territoire (SRAT) : C'est un plan qui s'occupe à la fois de distribuer les activités et le peuplement, et de localiser les infrastructures et les équipements à l'échelle régional (inter-wilayate). Le SRAT a pour objectifs : maîtriser la croissance des agglomérations, rationaliser l'exploitation des ressources naturelles, protéger et mettre en valeur le patrimoine culturel, les terres agricoles et les espaces ruraux, développer l'activité l'économie et l'emploi ;
- Les plans d'aménagement de wilaya (PAW) : Cet instrument agit à l'échelle de la wilaya dans les limites administratives, définit clairement les orientations majeures issues du SNAT et SRAT au profit des communes afin d'élaborer des plans directeurs d'aménagement et d'urbanisme et concrétiser les actions et les dispositions par des programmes communaux et/ou intercommunaux. Il permet également d'identifier la hiérarchie urbaine, fixer les vocations de chaque commune, et localiser les infrastructures, en assurant la distribution des activités et des services locaux d'intérêt public à travers le territoire de la wilaya [[Lakhdar Hamina and Abbas \(2015\)](#)] ;
- Plan communal de développement (PCD) : initié par le 2<sup>ème</sup> plan quadriennal (1974-1977), cet instrument annuel s'attache à la planification économique, sociale et spatiale au niveau local et principalement dans les villes de taille moins importante. Conformément aux dispositions de la loi n° 11-10 du 22 juin 2011 relative à la Commune, notamment l'article 107, le PCD constitue l'ensemble des opérations d'équipement et d'investissement programmées par l'Assemblée Populaire Communale (APC). Cet outil vise à fixer les populations, créer un équilibre territorial sans disparités, rationaliser l'occupation des sols, valoriser les ressources locales afin d'améliorer les conditions de vie des citoyens [[Chebli \(2011\)](#)] ;

- Plan sectoriel de développement (PSD) : Il constitue l'ensemble des opérations destinées au développement local sous formes de programmes répartis en secteurs d'activité et dirigés par différents directions et services de l'état chargés au niveau de la wilaya et de la commune ;
- Plan de modernisation urbaine (PMU) : le PMU gère les programmes de développement et repartit les investissements. Il concerne les villes plus ou moins importantes ou à développement rapide. Le PSD et le PMU permettaient un développement rapide, mais ils ont conduit à une consommation excessive foncière et financière, d'où la réflexion à d'autres instruments pour gérer l'espace urbain et les programmes d'équipement et d'investissement, c'est le plan d'urbanisme directeur (PUD) [[Lakhdar Hamina and Abbas \(2015\)](#)] ;
- Le plan d'urbanisme directeur (PUD) : c'est un instrument de base de la planification urbaine doté des moyens juridiques et techniques. Dans le cadre de son aménagement et développement à long terme chaque commune est dotée d'un PUD (l'ordonnance n° 67.24 du 18 Janvier 1967 et l'ordonnance N° 74.26, portant constitution des réserves foncières). Grâce au PUD, les autorités prennent des décisions objectives en matière de planification et développement des agglomérations du territoire en proposant l'organisation des activités, de l'habitat, des équipements, des transports sur le sol. Le schéma structurel concernant le déplacement temporaire d'une agglomération s'inscrit dans un plan d'urbanisme provisoire (PUP) qui précède l'élaboration du PUD ;
- Plan directeur d'aménagement et d'urbanisme (PDAU) dont l'objectif est de fixer les orientations de l'aménagement du territoire et d'urbanisme afin de définir les actions et les règles de l'utilisation des sols pour le territoire communal à court, à moyen, et à long terme. Il sert à élaborer les plans d'occupation des sols (POS) ;
- Plans d'occupation des sols (POS) : il précise les détails et les droits d'utilisation des sols et de construction à court et à moyen terme. De ce fait, toute implantation de lotissement ou de construction doit se référer au POS pour délivrance des autorisations lotir et permis de construction.

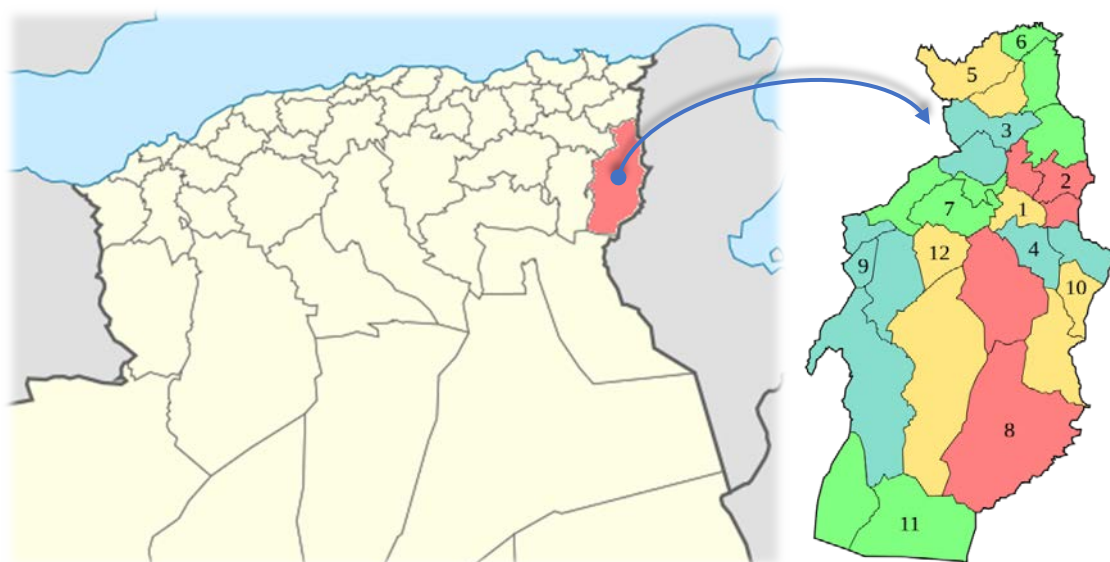
Les plans d'urbanisme PDAU et POS s'occupent d'étudier et de délimiter les périmètres des zones et terrains urbains y compris les terrains exposés aux risques naturels et industriels. Ils trouvent leur application à partir de 1991 suite au décret exécutif n° 91/178 du 28/05/1991 fixant les procédures d'élaborations et d'approbations des POS ainsi que le contenu des documents y afférents modifié et complété par le décret exécutif n° 05/318 du 10/09/2005. Et le décret exécutif n° 91-177 du 28 Mai 1991, modifié et complété par le décret exécutif n° 05-317 du 10/09/2005 fixant les procédures d'élaboration et d'approbation du plan directeur d'aménagement et d'urbanisme et le contenu des documents y afférents.

La loi 90-29 et les décrets d'exécution 90-175,90-176,90-177,90-178 du 28 mai 1990 ont permis d'établir une hiérarchie globale d'une planification descendante de SNAT au SRAT, puis de SRAT au PAW puis de PAW au PDAU et de PDAU au POS.

En effet, le modèle urbain en Algérie s'est développé comme la synergie des plusieurs modèles et expériences urbaines, et vu les liens historiques le modèle urbanistique algérien s'inspire en majorité du modèle français. Cette diversité a développé une richesse du patrimoine urbain, néanmoins la construction illicite dévastant le territoire urbain et l'exode rural ont conduit à une situation urbaine déstabilisée.

### *1.2.2. Tébessa : patrimoine urbain, particularités et perspectives*

Située dans les hauts plateaux à l'extrême est de l'Algérie, la wilaya de Tébessa incarne un patrimoine antique, mondialement classé et riche de sites historiques qui font sa particularité et témoignent la succession de grandes civilisations (byzantine, romaine, musulmane, et ottomane). Divisée en 28 communes réparties en 12 daïras, Tébessa s'étend sur une superficie de 13.878km<sup>2</sup> d'une population estimée à 768 188 habitants (fin de 2018) avec une densité moyenne de l'ordre de 53 habitants/km<sup>2</sup> dont 79,29% agglomérée chef-lieu, 16,44% en zones éparses, et 4,26% agglomérée secondaire. La wilaya de Tébessa est délimitée au nord par la wilaya de Souk-Ahras, à l'est les frontières tunisiennes, à l'ouest par les wilayate de Khenchela et d'Oum El Bouaghi, et au sud la wilaya d'El Oued<sup>2</sup>.



**Figure 1.1** – Situation géographique et administrative de la wilaya de Tébessa

Partant de ces composantes du patrimoine urbain et intégrant l'ensemble des potentialités offertes (ressources naturelles et minières, potentiel agricole, activités industrielle, sites touristiques, aéroport, établissements universitaires) permettant à Tébessa d'occuper une place importante dans la région. Cela impliquerait à la fois la valorisation de ce patrimoine comme un facteur essentiel de développement et l'élaboration d'une bonne stratégie de planification urbaine qui devra s'appuyer sur les composantes urbaines en impliquant l'ensemble des acteurs concernés.

Le tissu urbain de la wilaya de Tébessa s'est formé en agglomérations autour des anciennes cités (Bir El Ater, Cheréa, Ouenza, El Kouif, Ferkane) et des ressources

<sup>2</sup> Monographie de la Wilaya de Tébessa 2018 (Direction de la Programmation et du Suivi Budgétaire)

agricoles, minières et énergétiques. L'accroissement rapide et anarchique qu'ont connu ces villes au-delà de leurs capacités urbaines a engendré des tissus urbains peu structurés et faiblement équipés posant aujourd'hui de gros problèmes de gestion urbaine. Cette situation est due aux points suivants :

- Problèmes de viabilisation causés par la prédominance de l'habitat individuel illicite ;
- Vieux bâti et conditions de vie précaires dans des quartiers populaires d'une structure anarchique dépourvus de minimum de commodités ;
- La vétusté des réseaux d'assainissement, avec un sous dimensionnement des conduites dans les quartiers d'habitations illicites et l'absence d'épuration des eaux usées ;
- Un mauvais état du réseau de voirie, non revêtu, non aménagé, notamment dans les anciens quartiers ;
- Mauvaise répartition des infrastructures de base sur le territoire de la wilaya notamment les logements, le réseau routier, les établissements de santé, les établissements scolaires, les structures sportives et de loisir, les locaux administratifs et économiques ;
- Insuffisance des programmes en cours devant satisfaire les besoins des citoyens malgré les montants colossaux déployés.

L'enjeu essentiel consiste désormais à rétablir nos villes dans leurs dimensions fonctionnelles, en adoptant une bonne stratégie de développement permettant de mettre en place un système urbain hiérarchisé et articulé, portant sur des actions étudiées et réfléchies qui contribuent à l'équilibrage du territoire, à la mise en valeur et à la protection du patrimoine urbain de la wilaya. Il est donc primordial de réunir les conditions favorables pour les hisser au rang des villes durables.

Dans le cadre du programme quinquennal 2015-2019, la wilaya de Tébessa a envisagé la mise en œuvre des différentes opérations de structuration et d'aménagements urbains<sup>3</sup>, il s'agit de :

- La création de villes nouvelles et de centres à promouvoir, notamment au niveau de la bande frontalière et autour des principales agglomérations, qui serviront de support d'urbanisation ;
- La réalisation des infrastructures (administratives, scolaires, sportives, sanitaires, et de transport) nécessaires au développement local ;
- La réhabilitation et le développement du vieux bâti, constituant l'ancien tissu urbain, notamment dans la ville de Tébessa qui doit accueillir des infrastructures d'hôtellerie, d'affaires et autres, à l'image de son rôle dans la dynamique de développement régional ;
- La réhabilitation des grands ensembles urbains et la résorption de l'habitat précaire et insalubre ;

---

<sup>3</sup> Programme quinquennal 2015/2019 de la Wilaya de Tébessa (Direction de la programmation et du suivi budgétaires)



- La restauration et réaffectation des immeubles abandonnés en vue de les réintégrer dans le fonctionnement urbain ;
- La mise à niveau des équipements et services de base (raccordement aux réseaux AEP et assainissement, éducation, santé, sports, culture, etc.) ;
- La structuration du tissu urbain et l'aménagement des espaces publics par des travaux d'amélioration urbains ;
- L'amélioration des liaisons inter-quartiers tant au niveau de l'infrastructure que sur le plan de la qualité des services.

Ainsi, grâce à une prospection, une coordination et une mise en œuvre des instruments d'urbanisme, la wilaya a pu mettre à la disposition une superficie urbanisable de 2504,68 hectares, pouvant accueillir jusqu'à 89.000 logements avec l'ensemble des équipements et services nécessaires.

En vue d'une requalification de son tissu urbain et afin de réduire la densité de population des agglomérations dans la ville, la wilaya de Tébessa commence progressivement à s'investir à la création de pôles urbains d'équilibre recommandés dans le cadre du schéma régional d'aménagement des hauts plateaux Est. Aussi ces actions sont soutenues par un rééquilibrage urbain au niveau rural pour diminuer l'exode rural vers les centres urbains.

### **1.3. Pourquoi la planification urbaine est-elle importante ?**

Que ce soit une petite cité ou une grande métropole, la planification urbaine est indispensable pour maintenir et développer une communauté. Un plan urbain, un plan de la ville ou bien un plan directeur est un guide pour l'avenir d'une communauté. Il permet non seulement d'avoir un aperçu sur une cité en termes de routes, style d'immeubles, parcs etc., mais aussi de montrer comment une cité pourra fonctionner dans l'avenir en tenant compte de l'emploi, du nombre d'habitants, et des moyens de transport pour se déplacer dans cette cité [[Gallez and Maksim \(2007\)](#)].

La planification urbaine est « *une partie et pratique de l'urbanisme dont l'objectif est de prévoir l'évolution de l'urbanisme dans le temps, à partir d'un diagnostic de la situation actuelle, les tendances constatées et le projet de ville souhaité* » [[Saidouni \(2001\)](#)].

La planification sert à trouver des solutions aux problèmes des cités et des villes concernant les composantes urbaines tels que l'occupation des sols, logement, transport. En fait, elle consiste à déterminer concrètement comment les plans urbains sont élaborés, quels sont les acteurs y impliqués et les raisons pour lesquelles ces plans ont été élaborés.



**Figure 1.2** – Principaux champs d'intervention de la planification urbaine

### ***1.3.1. Planifier pour une meilleure occupation des sols***

Pour une meilleure occupation des sols [Kaiser et al. (1995)], les collectivités locales se chargent de la préparation et de la concrétisation des plans des cités et des villes. En adoptant un plan directeur officiel qui est le type plus répandu des plans urbains, soulignant le futur de la communauté à long terme, et présentant les différentes composantes de la communauté tels que l'habitat, le transport.

Les communautés urbaines utilisent d'autres plans spécialisés comme les plans stratégiques de revitalisation pour une périphérie souhaitant des résultats à court terme. La croissance des agglomérations engendre également des plans permettant la coordination du développement des cités et des environs de la région métropolitaine [Gordon et al. (2009)].

Certaines agences et organismes de développement sont impliqués à créer des plans non officiels pour la communauté, par exemple une association de voisinage peut développer un plan de rénovation des anciens immeubles en partenariat avec les résidents, les investisseurs locaux, et les autorités locales afin de réaliser ces plans.

### ***1.3.2. Planifier pour la satisfaction des citoyens***

Les villes se diffèrent et changent de caractéristiques dans le temps par rapport à la croissance de la population, aux moyens de transport, et le niveau de développement économique...etc.

Une grande population préfère habiter la périphérie plutôt que de vivre dans les zones métropoles. Des plans urbains de la périphérie visent à accueillir plus de populations alors que les zones urbaines non peuplées nécessitent des plans urbains de revitalisation [Sénécal et al. (2016)] dans le but d'attirer d'autres citoyens et garder les anciens résidents.

En tenant compte de la composition de la communauté et à travers un plan urbain, la planification urbaine détermine ce qui marche convenablement et ce qui nécessite une amélioration.

Un plan urbain est un guide pour le futur de la communauté, qui a un impact aussi bien pour les citoyens que pour les propriétaires et les investisseurs, par conséquent impliquer tous ces acteurs permet d'obtenir un plan urbain idéal. Les citoyens ont l'opportunité d'exprimer leurs désirs concernant l'endroit où ils vivent ainsi que leurs visions de cet endroit dans l'avenir [Zenker and Rütter (2014)].

L'utilisation des techniques moderne de la planification urbaine encourage le public à faire part à ce processus à travers les moyens possibles comme les rencontres, l'action associative, les réseaux sociaux, ...etc.

En effet, une bonne planification urbaine cherche en premier à satisfaire le citoyen en lui offrant des commodités assurant un niveau de vie agréable [Rafieian et al. (2009)].

### ***1.3.3. Planifier pour changer l'image de la ville***

De nos jours, les villes supportent une population plus grande qu'auparavant, d'autant plus certaines villes souffrent surtout d'une dégradation sur le plan infrastructures et d'une insuffisance des services, d'où l'exode des citoyens vers d'autres citées plus peuplées et agréables.

En effet, les villes évoluent et changent d'aspect dans le temps en termes de population, d'infrastructures, d'occupation de sol, de développement économique, et d'environnement. Tout cela est dû à plusieurs facteurs comme le mode de gestion de la ville, la disponibilité de ressources, et parfois aux catastrophes naturelles. Par conséquent les défis à relever par la communauté se multiplient.

Pour remédier à ces problèmes, plusieurs mesures notamment la revitalisation urbaine, et le redéveloppement local sont prises touchant en particulier l'exode urbain, les vieilles cités avec une qualité de vie déplorable pour les résidents, les mauvaises conditions urbaines des périphéries.

Ces mesures se concrétisent par l'élaboration de plans urbains qui répondent aux attentes des habitants et aident la ville à bien paraître et à bien fonctionner [Capel (1975), Rode (2017)]. Ils doivent tout de même prendre en charge tout éventuel changement.

### ***1.3.4. Planifier pour un développement économique***

Dans une société, le territoire urbain se caractérise par l'organisation spatiale de l'activité économique productrice de biens ou de services contribuant au fonctionnement social. Ces activités couvrent tous les domaines tels que la santé, le transport, la culture, le commerce, la communication.

L'étude socio-économique du territoire permet de connaître les potentialités dans les différents secteurs, et contribue ainsi à une bonne planification urbaine [Ammi (2019)]. Toutefois la mise en place d'infrastructures et programmes d'investissement avec une

couverture spatiale satisfaisante de l'ensemble du territoire assure un bon développement social, culturel et économique.

Cette planification tend essentiellement à augmenter la dynamique de la croissance économique et à améliorer le bien-être d'une société.

### ***1.3.5. Planifier pour protéger l'environnement***

Des villes bien planifiées veillent au respect et à la protection de l'environnement [Salles (2006)], par ailleurs les collectivités locales œuvrent pour élaborer des plans qui mettent en valeur la question de l'environnement qui traite les problèmes de la pollution de l'air, de l'eau et du sol.

Les excès de l'urbanisation et de l'industrie entraînent des changements sociaux et économiques de la ville, qui se répercutent sur l'environnement par un effet néfaste pour l'être humain et d'une exploitation excessive des ressources naturelles.

Etant toujours aux services du développement économique, la planification urbaine vise à mieux répartir les infrastructures et projets d'investissement sur l'ensemble du territoire en prenant les mesures nécessaires assurant à la fois un progrès économique et écologique [Beatley et al. (1997)]. Donc les plans urbains doivent ainsi comporter les moyens appropriés pour améliorer l'état environnemental sans compromettre l'avenir des générations futures.

## **1.4. Organisation du territoire urbain**

Le territoire urbain n'est pas seulement une mosaïque de zones différentes placées les unes à côté des autres, mais un système interdépendant où chaque zone ne fonctionne qu'en rapport avec les autres [Dubois-Maury and Paquot (2010)] de même la stratégie de développement local considère le territoire urbain comme étant une entité à part entière distinguée par ses propres caractéristiques où les décideurs et planificateurs auront besoin de recourir au partitionnement en zones urbaines « zoning » afin de mieux gérer ses composantes et contourner les problèmes relatifs. Plusieurs facteurs entrent en jeu pour définir la stratégie adéquate pour un découpage urbain notamment les facteurs administratifs et socio-économiques.

### ***1.4.1. Caractéristiques des zones urbaines***

La tâche de planification s'articule essentiellement sur le principe du « zoning » [Clément and Guth (1995)], qui facilite l'analyse de la zone urbaine en traitant ses propriétés et ses potentialités, dans le but de procéder à une gestion globale de tout le territoire urbain. La maîtrise du partitionnement assure la fiabilité et la cohérence des données relatives aux zones permettant d'une part aux planificateurs de produire des plans détaillés et consistants et d'autre part aux décideurs de bien les exploiter.

La prédominance de certaines caractéristiques dans une zone urbaine telles que la position géographique, la superficie, la population, les ressources naturelles, et le potentiel économique détermine les priorités à considérer dans l'élaboration d'un plan de

développement urbain, par exemple une zone caractérisée par une croissance économique suscite par conséquent l'extension de son réseau de transport, par ailleurs une zone avec une croissance démographique importante fait appel à revoir la politique de logement. Au-delà de ces caractéristiques, la notion de zone s'articule principalement sur les échelles : spatiale, administrative et sociale.

#### ***1.4.2. L'échelle spatiale***

Ici, le découpage géographique délimite les zones urbaines en fonction de leurs caractéristiques spatiales définies particulièrement par la position, les reliefs, et les limites géographiques. On y trouve les cités et les villes qui s'étendent sur tout le territoire urbain montrant visiblement les contours des agglomérations.

Ce découpage permet une planification urbaine concrète menant à un système d'information géographique (SIG) [Jenzer (1996), Pornon (2015)] offrant les outils nécessaires à la création, la manipulation, la visualisation, et la mise à jour des plans urbains.

#### ***1.4.3. L'échelle administrative***

L'organisation d'un système urbain obéit à un découpage administratif [Belhédi (1989)] fondée sur les centres d'agglomération, les périphéries, les cités, les villes, les métropoles voire même les régions.

Pour la création d'un système urbain hiérarchisé de qualité, Les autorités des différents niveaux administratifs - formant l'ossature du territoire urbain - assurent la gouvernance et mettent en place les mécanismes de gestion appropriés devant répondre aux enjeux et objectifs de développement.

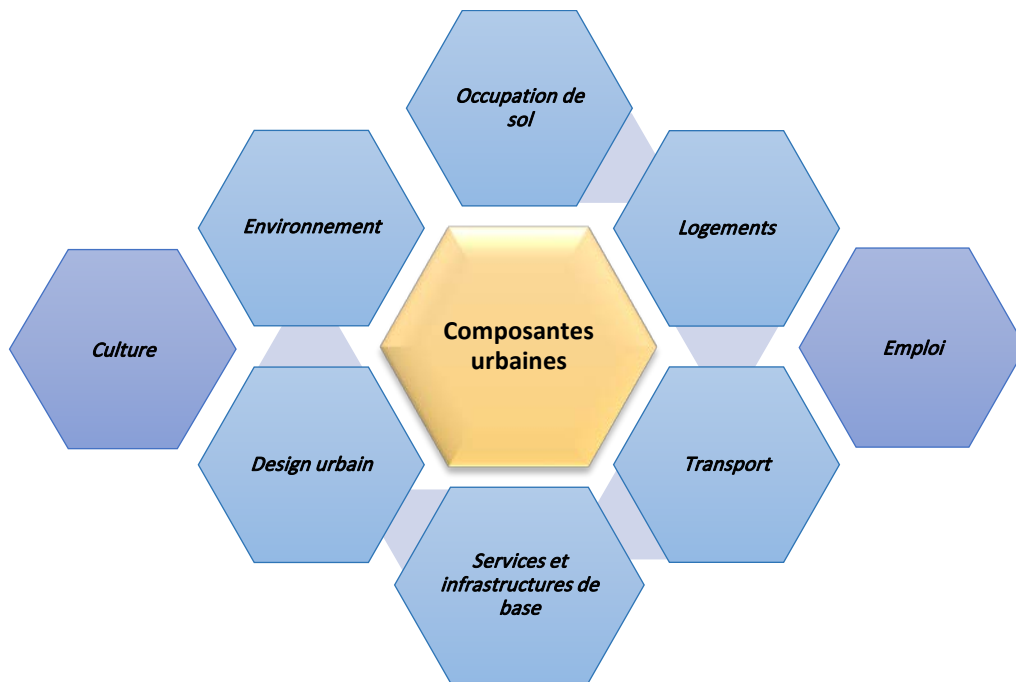
Les autorités habilitées se réfèrent constamment au découpage géographique pour une coordination entre les agglomérations adjacentes, et afin d'assurer la gestion administrative et économique.

#### ***1.4.4. L'échelle sociale***

Les approches traditionnelles en particulier celles de l'urbanisme rationaliste [Montedoro and Costa (2004)] donnent peu d'importance aux spécificités sociales et territoriales mais grâce aux nouveaux instruments d'intervention urbaine qui ont pour point commun de lutter contre les inégalités sociales et la fragmentation du territoire, il est possible de maîtriser l'évolution urbaine et d'en limiter les effets négatifs sur toute la société [Rabinovich and Navez-Bouchanine (2005)].

### **1.5. Les composantes de base d'un plan urbain**

Un plan urbain représente un schéma de développement de la ville à long terme, qui définit en quelque sorte l'aspect global de cette ville dans l'avenir. Mais il pourrait être compliqué quand il s'agit des différentes composantes de la ville y compris l'occupation des sols, le logement, l'éducation, la santé et l'hygiène, le transport, le sport et loisir, l'emploi, la culture, et l'environnement [Alexander (1992)].



**Figure 1.3** – Composantes de base d'un territoire urbain

Les zones urbaines sont différentes car chaque zone possède ses composantes spécifiques qui font partie d'un plan urbain. D'où l'intérêt économique est très important dans certaines zones alors que dans d'autres zones la priorité est donnée à la protection des ressources naturelles.

Les plans urbains prennent en considération les détails de toutes les composantes et comment ces derniers sont-ils reliés entre eux et fonctionnent en harmonie, par exemple l'impact des moyens de transport ou bien l'activité industrielle sur l'environnement. Ainsi, la coordination s'avère indispensable pour une bonne planification.

Ces plans doivent être adaptés aux exigences de la communauté en offrant un design efficace et esthétique en respectant les contraintes : sociales, économiques et environnementales.

#### ***1.5.1. Occupation de sol***

L'occupation de sol [Kaiser et al. (1995)] est l'élément le plus important dans un plan urbain qui désigne l'endroit où chaque composante doit être situé. Donc, chaque assiette peut être réservée pour accueillir des infrastructures à usage résidentiel, commercial ou bien industriel.

Le recensement de tous les terrains d'une zone urbaine permet d'avoir un aperçu sur la manière dont ces terrains sont occupés, y compris d'autres caractéristiques comme les autoroutes, les chemins de fer, les rivières. Le modèle futur d'utilisation des terrains est illustré par des cartes montrant comment ces terrains seront exploités rationnellement par la communauté en s'assurant que l'ensemble des composantes fonctionnent de manière cohérente et durable.

### ***1.5.2. Logements***

Le logement représente le noyau de la planification urbaine [Legrand (1992)], donc un plan urbain doit prendre en compte la quantité et qualité du logement dans une communauté urbaine. L'évaluation des conditions de logement repose sur un éventail de caractéristiques, telles que le type de logement (individuel, appartement, ...etc.), l'âge et l'état du logement, ainsi que d'autres critères relatifs à la propriété et la commodité du logement.

A partir de la situation actuelle de l'habitat d'une zone urbaine et pour mieux servir la communauté en logements, il faut prévoir l'évolution de la population et de l'économie afin d'exprimer les besoins en termes de nombre et type de logements.

D'autres composantes comme l'occupation de sol et le transport peuvent entrer en jeu pour l'élaboration d'un plan futur du logement qui comporte à la fois le choix des terrains à promouvoir et les logements existants nécessitant une réhabilitation.

### ***1.5.3. Transport***

Cette composante concerne les moyens permettant de relier une métropole par des chemins piétons, pistes cyclables, des routes, des chemins ferroviaires, des rivières, ...etc.

L'évaluation des conditions de transport dans cette ville s'appuie sur la facilité de se déplacer d'un lieu à autre et à quel point les citoyens sont servis.

Une planification de transport [Offner (2002)] nécessite d'une part l'identification du système de transport existant en améliorant la gestion de ce système et d'autre part proposer des améliorations futures.

L'expansion du réseau du transport pourrait avoir un impact sur les autres composantes urbaines tels que l'occupation de sol, et l'environnement. Il est donc préférable d'opter pour d'autres alternatives comme le transport en commun, les vélos.

### ***1.5.4. Infrastructures de base et services***

Certaines infrastructures d'une ville, telles que les routes et les lignes électriques, sont faciles à voir. Mais un simple citoyen ne donne pas assez d'importance à ce qui fait fonctionner la ville, comme les conduites souterraines, les centrales électriques, et les stations d'épuration.

Les villes fournissent également à la communauté de nombreux services, tels que : les écoles, les hôpitaux, les structures sportives et de loisir, la police et la protection civile, qui sont abordées dans les plans urbains et leur planification est étroitement coordonnée avec les modèles d'occupation de sols de la ville afin de déterminer où il convient de situer et de fournir l'infrastructure nécessaires et ses services.

Planifier des améliorations futures à apporter aux infrastructures et aux services est très important car les éléments tels que, les écoles, les centrales électriques, les réseaux

d'assainissement qui assurent un service pendant des décennies sont très coûteux [Breuillé et al. (2019)]. Malgré les coûts élevés pour des changements durables, une planification réfléchie assure que les villes puissent fournir des services essentiels dans l'avenir.

#### *1.5.5. Design urbain*

Le design urbain « conception urbaine » d'un plan urbain traite les deux côtés esthétiques et fonctionnels [Paetzold and Rollet (2011)]. Toute ville est constituée de nombreux immeubles, y compris des résidences individuelles, des bâtiments, des rues et des quartiers. La conception de chacun de ces types de structures contribue à la fois à la beauté et à l'utilité de la ville. Par exemple, une rue bien conçue est une rue attrayante qui permet aux véhicules de circuler librement et qui possède de bons trottoirs où les gens peuvent se balader.

En examinant le design des zones existantes et leurs caractéristiques, le plan urbain doit fournir des suggestions pour la conception d'un développement futur. De manière à améliorer son attrait esthétique et sa fonctionnalité, le plan du design urbain doit assurer la coordination avec les autres composantes, par des directives de conception pour un développement futur des logements qui offrent un aspect esthétique et convivial.

La conception urbaine d'un plan implique souvent la participation du public en collectant les idées et les opinions de la communauté à travers des esquisses de conception, ce qui aide les parties prenantes à avoir une vision de la communauté et comment elle fonctionne dans l'avenir.

#### *1.5.6. Environnement*

Les plans urbains doivent traiter la question de l'environnement [Paquot and Younès (2010)] en tenant en compte la préservation des ressources naturelles, de la faune, de la flore, et de la qualité de l'air, auxquels s'ajoute la création d'espaces ouverts.

En coordination avec l'ensemble des composantes auxquelles il est relié, tels que le transport, les infrastructures et l'occupation de sol, la composante environnement d'un plan urbain évalue l'état général de la ville et examine également l'influence de la communauté sur l'environnement en particulier le facteur pollution causée par les ménages, les usines, et les véhicules provoquant la dégradation de l'environnement et des effets néfastes sur la santé.

Afin de minimiser l'impact global sur l'environnement, il s'agit de réduire les niveaux locaux d'émissions de gaz à effet de serre qui contribuent au changement climatique mondial.

### **1.6. Le projet urbain et aménagement des villes**

Depuis un demi-siècle, le projet urbain demeure un concept évoquant un contexte socio-économique en évolution, qui est largement employé dans le domaine d'aménagement



comme étant une alternative à la planification urbaine issue de l'urbanisme rationaliste [Courcier (2005), Rabinovich and Navez-Bouchanine (2005)].

Le projet urbain est polysémique, dont les définitions dépendent essentiellement de son emploi par les différents usagers tels que les autorités locales, les architectes-urbanistes, les aménageurs, et les entreprises.

### 1.6.1. Qu'est-ce qu'un projet urbain ?

Le projet urbain, projet de ville ou projet d'agglomération [Ascher (1991)] fait appel aux acteurs qui doivent non seulement contribuer à la réflexion et à la consultation mais aussi au processus d'élaboration du projet [Ascher (1995)]. A travers une démarche de consultation et de partenariat avec l'ensemble des acteurs, les projets consensuels nécessitent d'identifier les potentialités, les problèmes de la ville, et les enjeux majeurs afin d'atteindre les objectifs de la planification stratégique.

De même, un projet pour la ville constitue une image collective dessinée par les décideurs en collaboration avec la population autour d'une conception partagée d'un avenir commun [Merlin (1996)].

Trois décennies après l'apparition de cette notion qui a renouvelé les principes d'aménagement des villes, Ingallina [Ingallina (2001)] souligne qu'il « s'agit d'un concept et d'une manière d'agir en formation qui marquent un moment de transition entre la manière traditionnelle de penser l'urbanisme et une nouvelle approche, moins figée et plus ouverte aux transformations et aux débats ».

L'essentiel de cette notion soutiré par Ingallina en quelques aspects majeurs :

*« ...un projet urbain ne se fait pas en un jour, mais il accompagne le processus de transformation urbaine dans la durée ; il ne peut pas répondre à la logique de l'urgence souvent invoquée par les maires. Il doit réunir des compétences multiples, car il s'applique à la ville qui est une réalité complexe, pas unique où formes matérielles et formes sociales sont liées dans des relations qui se sont établies dans le temps et dont il devra rendre compte. Il se réfère à une multiplicité de techniques dont la maîtrise ne peut être confiée aux seuls architectes ou ingénieurs, mais demande, selon le cas, d'autres compétences spécifiques et nécessaires pour sa faisabilité (y compris financière). Puisqu'il a une visée large, il doit permettre le débat et l'échange avec la population dont l'avis est déterminant. »*

On pourrait constater que la notion de projet urbain suit sa visée et dépend du contexte auquel il est rattaché en s'échappant aux règles strictes de la planification classique.

Dans un contexte général, il désigne « une pratique planificatrice ouverte et souple » [Courcier (2005)] pour se libérer des méthodes de planification traditionnelles dépassées par rapport aux problématiques d'aménagement et aux exigences des acteurs impliqués, donnant naissance notamment à l'approche morphologique qui s'oppose à une planification jugée rigide.

La morphologie des villes comporte des valeurs historiques et archéologiques, c'est à partir de l'analyse de la cartographie et de l'historique des différents tissus urbains, que des informations sont extraites et consolidées permettant d'orienter les opérations urbaines [Bouchareb (2011)].

Le projet urbain possède une dimension spatio-temporelle, qui définit ses composantes par rapport à l'espace urbain dans le temps, où chaque acteur agit dans la fabrication de la ville. En d'autres termes le projet urbain comporte la projection des activités et aménagements urbains à entreprendre aussi bien dans l'espace que dans le temps. Ces activités subissent des éventuelles révisions afin de mieux servir la communauté.

En effet, la démarche du projet urbain s'attache aux pratiques sociales du passé et à l'histoire de la ville, en dictant et en suggérant aux acteurs les directions à suivre dans l'avenir.

Malgré sa polysémie, la notion générique du projet urbain réside en son pouvoir de créer une liaison entre le passé et le futur dans une vision planificatrice ouverte, souple et réflexive. Ainsi la réussite des projets urbains nécessite l'adhésion collective, la concertation, la participation de tous les acteurs concernés de loin ou de près y compris les habitants de la ville.

### *1.6.2. Plan urbain et développement durable*

Vers une stratégie de coordination des actions, en disposant des moyens et des ressources nécessaires, le plan urbain est un raisonnement linéaire guidé par la relation objectifs/contraintes, qui joignent les idées aux actions pour assurer un ordonnancement et un équilibre urbains en incitant les acteurs à y participer positivement.

D'autant plus, le plan urbain accorde un privilège particulier à la vie communautaire, par la mise à disposition des espaces publics (jardins, aires des jeux, espaces verts). La réhabilitation et la revalorisation des lieux abandonnés [Gandreau et al. (2018)] permet de gagner des espaces et de les rendre vivables.

Afin d'atteindre les objectifs visés, il faudra élaborer soigneusement un plan permettant la gestion et la préservation des ressources, notamment les ressources naturelles, foncières et financières, en donnant un grand intérêt au développement social, économique, culturel et environnemental.

Les dynamiques urbaines (étalement, mitage<sup>4</sup> des espaces, segmentation sociale), les comportements de mobilité, les pratiques alimentaires (associations pour le maintien de l'agriculture paysanne, jardins partagés...), et les attitudes énergétiques (équipements individuels de modes alternatifs de production) sont remis en cause pour leurs impacts environnementaux [Dubois-Maury and Paquot (2010)].

---

<sup>4</sup> En urbanisme, à l'instar des mites, le mitage est une conséquence de l'étalement urbain. Il désigne l'implantation d'édifices dispersés dans un paysage naturel.

En effet, l'environnement constitue un enjeu majeur et une préoccupation individuelle, d'implication sociale, et de mobilisation associative. En lien à l'environnement, le développement durable est l'un des principaux défis (ville durable, écoquartiers, transition socio-écologique...) auxquels sont confrontées les collectivités locales.

Grâce aux approches de planification promues par l'environnement, ou par la ville durable (projets de quartiers durables, actions de protection des centres historiques, et politiques énergétiques) [Mathieu and Guermond (2005), Emelianoff (2007)], la gouvernance urbaine se développe en exploitant des alternatives telles que les énergies renouvelables, l'économie verte, le recyclage, etc.

Enfin, l'avènement des nouvelles technologies ont permis aux systèmes de développement urbain de fonctionner de manière efficace et durable, car celles-ci contribuent activement à la gestion optimale du territoire et à la préservation des ressources. Tirant profit de ces technologies, les cités deviennent intelligentes « smart city » et capables de combler les déficits et garantir une dynamique dans différents secteurs d'activité [Su et al. (2011)], afin d'assurer un bien-être et une qualité de vie meilleure des citoyens.

### ***1.6.3. La démarche de projet urbain***

Le projet urbain [Mangin and Panerai (1999)] constitue une révolution dans l'urbanisme de planification opposant les approches sectorielles. Il agit tant à l'échelle locale qu'à l'échelle territoriale, en invitant tous les intervenants jugés potentiels de manière à leur déléguer le pouvoir décisionnel [Toussaint and Zimmermann (1998)]. Cette démarche urbaine permet de rapprocher les idées afin de trouver des compromis qui réunissent les acteurs dont les opinions et les intérêts sont différents voire même conflictuels.

Le fait de remettre en cause la planification hiérarchique, a mis fin à la dominance du pouvoir centralisé en offrant plus d'autorité et de liberté d'actions pour les collectivités locales [Epstein (2013)]. En effet, la mise en scène du projet urbain nécessite en premier de vouloir changer positivement le territoire sur tous les plans (économique, social, culturel, environnemental, etc.) à partir des inspirations et des convictions [Bouchareb (2011)]. En employant les stratégies adéquates, cette volonté s'accroît par un minimum requis d'actions à accomplir dans le temps par rapport aux perspectives auxquelles les acteurs qualifiés doivent coopérer simultanément.

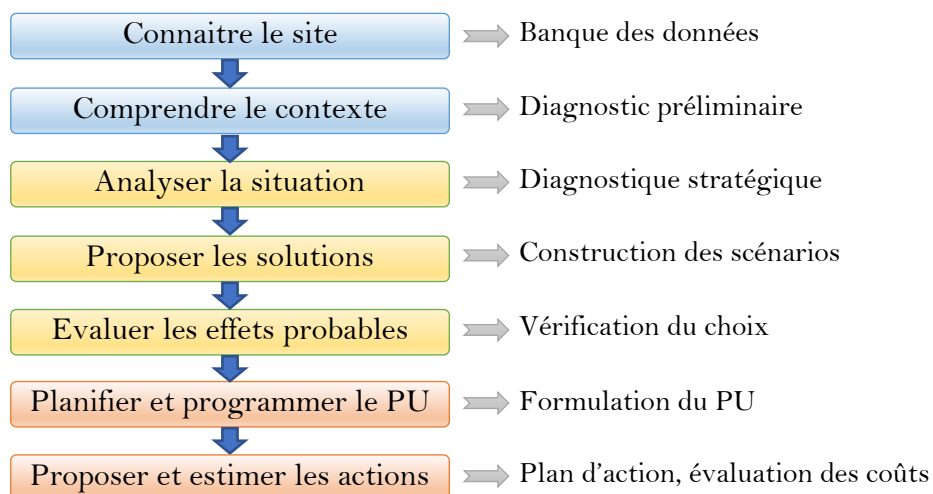
Généralement la mise en œuvre d'un projet urbain s'appuie sur l'adhésion de plusieurs intervenants notamment les autorités locales, les bailleurs de fonds, les compétences techniques et scientifiques, les usagers, et les utilisateurs, d'où un comité de pilotage est désigné, appelé chef de projet ou *maître d'ouvrage*.

Conformément à la politique urbaine, la maîtrise d'ouvrage est assurée au niveau territorial par le choix de la stratégie de planification adéquate, qui dessine l'image de la ville à long terme. Le *comité technique* s'occupe de la coordination entre les différents acteurs, quant aux *porteurs de projets* (collectivités locales et territoriales, propriétaires fonciers publics et privés, investisseurs, etc.) ils constituent le centre de décision et

d'orientation de la démarche de projet urbain. Ainsi, les opérations de projet urbain s'exaucent grâce à la participation des professionnels formant la catégorie *maitre d'œuvre*.

La réussite du projet urbain sollicite également la consultation et l'intervention des usagers qui poursuivent des intérêts et en subissent les conséquences. Cette catégorie sociale représente souvent les habitants et les associations locales dont les attentes devant apparaître explicitement dans les programmes, en appliquant même des pressions sur la démarche de projet afin d'atteindre leurs objectifs. Cependant, ces programmes comportent un éventail d'opérations à entreprendre dans divers secteurs d'activités au niveau territorial, et ce en fonction des ressources et outils disponibles.

La démarche du projet urbain se déroule à travers les étapes suivantes : le montage, la programmation, la conception et la réalisation. Le *montage* constitue l'étape initiale qui précède le lancement du projet et consiste à : définir les objectifs, mobiliser les acteurs concernés, évaluer les délais et les coûts. En concertation avec l'ensemble des acteurs, l'étape de *programmation* définit clairement les objectifs et les attentes afin de rédiger un cahier de charges. Quant à la *conception*, elle est confiée aux compétences techniques et scientifiques (architectes et urbanistes) pour formaliser le projet à partir du cahier de charges sous forme de plan d'urbanisme (plan de quartier, plan d'affectation, plan directeur, etc.). Finalement l'étape de *réalisation* concerne la mise en œuvre ou les travaux de construction en impliquant les acteurs locaux dans les tâches d'orientation et d'organisation. Pratiquement, ces étapes peuvent se superposer et ne suivent pas obligatoirement un ordre linéaire [Dind (2011)]. D'ailleurs, les étapes de la démarche du projet urbain pourraient être décortiqué autrement, comme pour le cas illustré dans la figure suivante (Figure 1.4) :



**Figure 1.4** – Démarche et étapes du projet urbain  
(Source : [Berezowska-Azzag et al. (2012)])

La démarche du projet urbain se montre d'une efficacité remarquable aussi bien sur le plan théorique que sur le plan pratique. Il est donc conseillé de renforcer cette démarche par l'intégration de nouvelles technologies de l'information permettant de suivre les changements contemporains et planifier la ville d'une meilleure façon.


### **1.7. Conclusion**

La complexité urbaine constitue une réalité qui évolue constamment que les instruments classiques d'urbanisme ont montrés leurs échecs pour contrecarrer les effets négatifs du processus urbanistique. Tant pour les grandes villes, que les petites villes, les problèmes varient selon les contextes géographiques, historiques, politiques, économiques, et sociaux de chacune. Par conséquent l'adoption de nouvelles méthodes, démarches et outils de planification et d'aide à la décision demeure un enjeu majeur.

Le projet urbain constitue une révolution dans l'urbanisme de planification opposant les approches sectorielles. Il agit tant à l'échelle locale qu'à l'échelle territoriale, il est désormais primordial d'impliquer l'ensemble des acteurs concernés par le processus de planification urbaine tels que les décideurs, les bailleurs de fond, les gestionnaires, les concepteurs, les promoteurs, les investisseurs, les intellectuels, les chercheurs, et les citoyens pour réunir les idées vers une vision globale de l'avenir de la ville.

Fortement liée au projet urbain, la planification urbaine permet aux pouvoirs publics et privés grâce aux instruments d'urbanisme de prévoir, d'orienter, d'organiser et de maîtriser le développement du territoire urbain. Cette pratique urbanistique s'articule sur l'analyse du territoire et le diagnostic de l'état présent de la communauté, qui conditionnent la prise de décision. Il faudrait bien comprendre le fonctionnement de la communauté urbaine et savoir à quel point elle tend vers le changement pour dresser de meilleurs plans qui assurent une bonne répartition des composantes urbaines.

Afin d'atteindre les objectifs souhaités par le projet urbain, le plan urbain projette à l'échelle temporelle les opérations urbaines à concrétiser sur le territoire en mettant en valeur l'histoire et le patrimoine de la ville sans pour autant estomper son identité.

 *Chapitre - 2*

# Les systèmes intelligents d'aide à la décision



# Chapitre 2 - LES SYSTEMES INTELLIGENTS D'AIDE A LA DECISION

## 2.1. Introduction

Les systèmes intelligents d'aide à la décision (SIAD) ont fortement attiré l'intérêt des chercheurs grâce à leur utilisation dans plusieurs domaines, notamment la santé, la veille économique (*business intelligence* BI), les industries, la fabrication et les systèmes de contrôle. Combinant l'intelligence artificielle (IA) et le système d'aide à la décision (SAD), les SIADs utilisent de façon intensive des techniques d'intelligence informatique « *Computational Intelligence* CI » pour créer des systèmes informatiques puissants.

Doté de capacités cognitives, le SIAD fournit un accès facile aux informations pertinentes et emploie un raisonnement similaire à celui de l'homme pour la prise de décision dans la résolution de problèmes. Il surmonte les limitations causées par des données ambiguës, incertaines et non structurées, qui sont souvent largement distribuées et provenant de sources différentes [Tweedale et al. (2016)].

Actuellement, le succès du SIAD dans divers domaines d'application motive la communauté de la planification urbaine à en tirer profit et particulièrement pour l'élaboration de projets urbains.

Ce chapitre introduit l'évolution des systèmes d'aide à la décision en mettant l'accent sur les différents systèmes intelligents d'aide à la décision et leurs domaines d'application.

## 2.2. Evolution des systèmes d'aide à la décision (SAD)

Les processus de décision que ce soient individuels ou collectifs s'appuient sur de nombreuses solutions technologiques en tant que systèmes d'aide à la décision (SAD) qui ont vu le jour au début des années 70. Celles-ci accompagnent de près l'évolution des technologies de l'information [Bohanec (2003), Power (2007)].

En se référant aux activités de management d'Anthony et à la description des décisions de Simon, Gory et Scott Morton [Shim et al. (2002)] qui ont mis en évidence le concept de SAD en utilisant les termes : structuré, non structuré et semi-structuré. Ils ont également utilisé la description du processus décisionnel de Simon (Intelligence, Design, Choix).

Le SAD est un système informatique traitant des problèmes contenant des parties non structurées ou semi-structurées, sur lesquelles repose le jugement du décideur [Shim et al. (2002)]. Les SAD de base comportent les connaissances du décideur humain, associées à des données pertinentes et aux fonctionnalités de l'interface utilisateur. Fondés sur des modèles quantitatifs informatisés, ils appliquent des méthodes mathématiques et statistiques pour suggérer ou recommander des actions aux managers [Power (2007)].

En s'appuyant sur les fonctionnalités dont les SAD réagissent, on peut les classer en trois principales catégories : SAD passifs, SAD actifs et SAD proactifs [Kwon et al. (2005)]. Le premier est un SAD intégrant des connaissances, dont les composants sont des entrepôts de données, des traitements analytiques en ligne *Online Analytical Processing* (OLAP) ainsi que des bases de règles, tandis que le second est doté des capacités d'apprentissage supplémentaires, notamment système expert, SAD adaptatif, Système à base de connaissances (SBC), ces derniers forment une partie des systèmes intelligents d'aide à la décision (SIAD). La troisième catégorie, également connue sous le nom de SAD basé sur l'informatique ubiquitaire, qui se caractérise par l'intégration de la sensibilité au contexte, la portabilité, la mobilité et la réactivité.

### **2.3. Le passage des SADs vers les SIADs**

Les technologies de l'information avancées ont donné naissance à un nouveau type de SAD appelé système intelligent d'aide à la décision (SIAD). Doté de l'intelligence artificielle, le SIAD intègre des fonctionnalités d'analyse, de prédiction et de raisonnement pour aider les décideurs à résoudre des problèmes complexes à partir des données non structurées, incertaines, ambiguës et/ou incomplètes. Ces systèmes intelligents sont largement utilisés dans une multitude de domaines, notamment la gestion, l'industrie, la santé et le commerce.

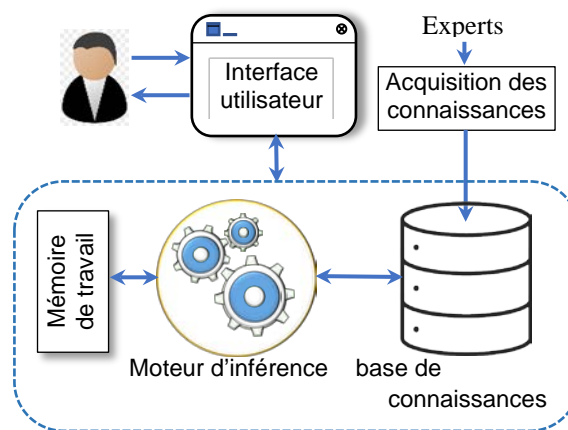
Grâce aux outils de l'intelligence artificielle, le SIAD réagit de façon similaire au raisonnement de l'être humain pour faire un choix et envisager un plan en fonction d'un objectif. Par exemple, dans l'urbanisme un SIAD combine les connaissances du domaine, avec les données relatives aux infrastructures, aux citoyens, et aux services afin d'assister les experts et les planificateurs à élaborer des plans urbains. Un tel système pourrait offrir des résultats précis, fiables et utiles [Cortés et al. (2000)].

Ces systèmes ont trouvé succès dans divers domaines d'application tels que la médecine, l'industrie, l'éducation, l'ingénierie, l'économie, les finances, l'armée, et la bio-informatique. Généralement les SIAD englobent les systèmes à base de connaissances, les réseaux de neurones, les systèmes d'aide à la décision assisté par des agents intelligents, les SAD basé sur les ensembles approximatifs, la fouille des données et la fouille des processus, l'apprentissage machine ainsi que d'autres systèmes basés sur les techniques d'optimisation tels que les systèmes flous, et les systèmes basés sur les algorithmes génétiques [Das (2016), Tweedale et al. (2016)]. Ces systèmes constituant le noyau des SIAD seront présentés dans ce qui suit.



### 2.3.1. Les systèmes à base de connaissances

Le système à base de connaissances (SBC) est l'un des systèmes intelligents d'aide à la décision les plus connus. Certainement, l'emploi des techniques de l'intelligence artificielle a permis de résoudre les problèmes complexes dont la résolution était impossible par les techniques usuelles. La combinaison des technologies de l'informatique telles que l'ingénierie des connaissances, le traitement de langage naturel, les systèmes d'aide à la décision, l'intelligence artificielle, les systèmes de gestion des bases de données, l'ingénierie des logiciels, la représentation des connaissances, le raisonnement à base de cas (RBC), les réseaux de neurones, et le web sémantique a produit ce qu'on appelle le système à base de connaissances qui a développé l'approche du système expert.



**Figure 2.1** – Architecture d'un système à base de connaissances

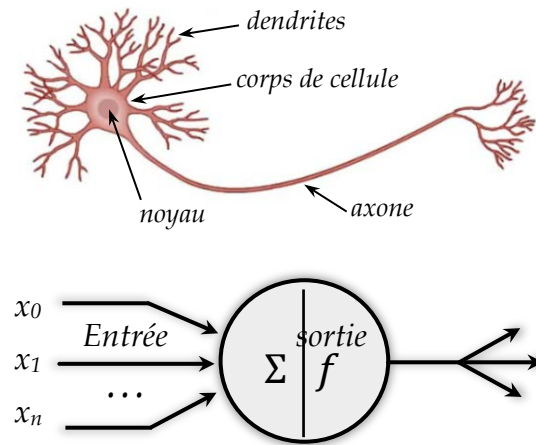
Les fonctions les plus puissantes d'un SBC consistent à raisonner et à générer de nouvelles connaissances, en utilisant trois composants principaux : la base de connaissances explicites, le moteur d'inférence et la mémoire de travail (Figure 2.1). Quelle que soit l'approche utilisée par le moteur d'inférence ; chaînage avant, chaînage arrière ou une toute autre stratégie, l'objectif est d'inférer de nouvelles connaissances en appliquant des propositions logiques. La connaissance est souvent représentée par des règles de la forme IF-THEN.

### 2.3.2. Les systèmes à base de réseaux de neurones

Les réseaux de neurones artificiels (RNA) sont inspirés du système nerveux humain, cette technique consiste à produire une machine qui imite le fonctionnement du cerveau de l'être humain et capable d'apprendre en utilisant la méthode *essai et erreur*.

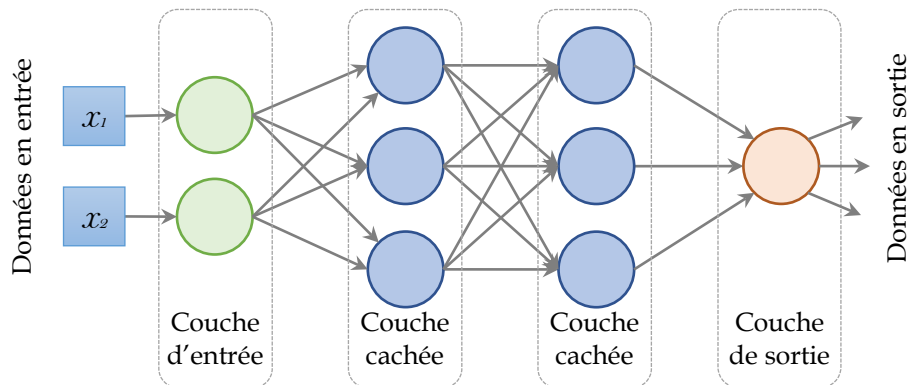
Les RNA se caractérisent par leur capacité d'apprendre à partir des données floues, déformées ou incomplètes [Glorfeld and Hardgrave (1996)].

Les blocs essentiels d'un RNA sont les neurones qui simulent les neurones tellement aux neurones biologiques par leurs corps et leurs branches (Figure 2.2). En fait, le réseau de neurones forme un chaînage d'arborescences de neurones qui agissent ensemble.



**Figure 2.2** – Perceptron biologique VS perceptron artificiel

Le RNA est conçu comme une séquence de couches (Figure 2.3). Chaque couche est formée de neurones appelés nœuds, qui sont connectés à ceux de la couche suivante par des synapses<sup>5</sup>. Chaque nœud reçoit des signaux d'entrée et répond par des signaux de sortie. Les valeurs en entrée sont véhiculées en tant que signaux de la couche d'entrée vers la couche de sortie, en passant par les couches de neurones intermédiaires appelées les couches cachées.



**Figure 2.3** – Les réseaux de neurones artificiels

Ce processus consiste essentiellement en une boucle de prédiction, ça fonctionne par ajustement du poids afin de minimiser les erreurs globales résultant du réseau qui se base sur les fonctions de transfert à partir de chaque nœud.

### ***2.3.3. Les systèmes assistés par les agents intelligents***

Etant un domaine de recherche très actif de l'IA, les agents intelligents [Etzioni and Weld (1994), Khoong (1995)] et les systèmes multi-agents (SMA) évoluent en parallèle avec les systèmes d'aide à la décision intelligents.

<sup>5</sup> Une synapse est une structure élémentaire ou bien l'unité fonctionnelle entre deux neurones, c'est le contact de brin d'axone d'un neurone avec une dendrite d'un autre neurone.

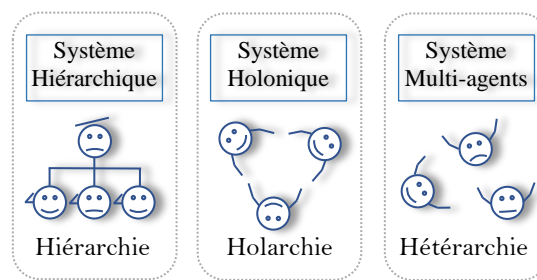
Dans un environnement réseau distribué ou centré ces agents partagent des ressources et interagissent avec d'autres agents par un échange de données, voire même par la coopération, la coordination et la négociation [Phillips-Wren (2008), Zaraté et al. (2008)]. Ils servent également d'assistants ou de consultants pour collecter des informations, restituer des données spécifiques, analyser des données, répondre aux requêtes des utilisateurs, planifier des tâches, ou bien générer des rapports de données [Tweedale et al. (2016)].

En fait, un agent est un système à base de logiciel informatique doté des caractéristiques suivantes [Jennings et al. (1998), Huhns and Singh (1998), Wooldridge and Jennings (1995), Padgham and Winikoff (2005), Zaraté et al. (2008), Phillips-Wren (2008), Wooldridge (2009), Chen and Ichalkaranje (2013), Kantamneni et al. (2015), Russell and Norvig (2016)] :

- **Autonomie** : les agents opèrent indépendamment sans l'intervention humaine.
- **Coopération** : les agents agissant en coopération avec d'autres agents pour atteindre les objectifs soulignés.
- **Réactivité** : les agents réagissent immédiatement à tout changement d'environnement.
- **Proactivité** : non seulement les agents agissent pour leur propre environnement, mais ils sont également capables de prendre des initiatives propres suivant les objectifs.
- **Sociabilité** : l'agent peut communiquer et interagir avec les autres agents (logiciels et humains) afin d'accomplir ses tâches et de les aider à compléter les leurs. Ces trois dernières caractéristiques définissent ce qu'on appelle la *flexibilité* d'un agent.
- **Rationalité** : le comportement cognitif permet à l'agent de choisir l'action appropriée pour atteindre son but par l'utilisation efficace des ressources.
- **Mobilité** : si c'est nécessaire, l'agent se déplace sur des réseaux informatiques pour transporter des connaissances et des données, ou pour effectuer d'autres tâches
- **Engagement** : l'agent coopératif s'engage à accomplir les actions qui satisferont son but, et met en place un plan d'actions en coordination avec d'autres agents.
- **Adaptation** : les agents ont la capacité d'apprendre et de modifier leur comportement en fonction de la base de connaissances, et selon l'agent avec qu'il interagit.
- **Apprentissage** : à partir des expériences passées, l'agent évolue et améliore ses connaissances, afin de s'adapter aux différentes situations.
- **Intelligence** : un agent intelligent est capable d'apprendre, raisonner et planifier ses actions en tenant compte de celles des autres agents. C'est un agent cognitif, rationnel, et adaptatif.
- **Orienté-objectif** : les agents ont leur objectif spécifique à accomplir.

- **Persistance** : les agents peuvent vivre et maintenir leur état le plus longtemps possible.

Les agents intelligents et les systèmes multi-agents ont un impact important sur le système d'aide à la décision qui couvre de nombreux domaines tels que les systèmes industriels et de fabrication, la gestion d'organisation, la médecine et l'enseignement électronique (e-learning). Ils simulent d'une certaine manière le fonctionnement des entités du monde réel et facilitent le processus décisionnel dans des situations complexes nécessitant de réagir rapidement.



**Figure 2.4** – Les architectures hiérarchique, holarchique, et hétérarchique des SMA (Inspiré de [Bakos and Dumitrascu (2017)], Source [Khelifa and Laouar (2020)] ).

Dans la littérature des SMA, l'organisation des agents présente trois architectures principales : la hiérarchie, l'holarchie et l'hétérarchie (Figure 2.4). Dans l'architecture classique, les agents sont conçus pour former un système hiérarchique, alors que l'holarchie faisant référence à des systèmes multi-agents holoniques, dans laquelle les agents sont des entités considérées à la fois comme des "tout" et des "parties" [Koestler (1967)]. Cependant, dans l'hétérarchie, les agents sont des éléments non ordonnés, ils sont des entités totalement autonomes et indépendantes [Bakos and Dumitrascu (2017)].

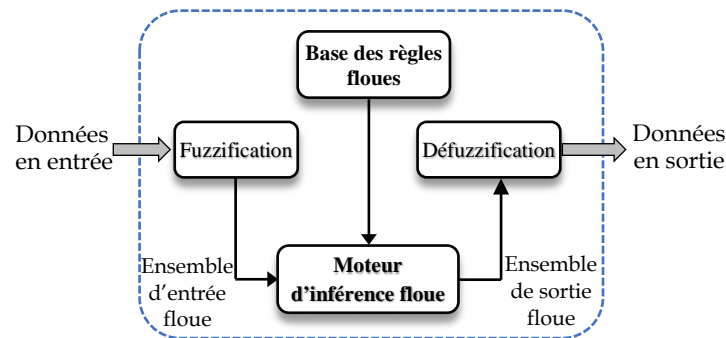
#### 2.3.4. Les systèmes basés sur la logique floue

Le système flou (SF) est une technologie qui a prouvé sa capacité à développer de nombreux types de systèmes décisionnels, particulièrement quand on manipule différentes formes de connaissances. Dans la pratique, la plupart des systèmes informatiques ne supportent pas les informations linguistiques dont la description nécessite une certaine expérience similaire à celle de l'être humain. Par conséquent, le concept de la logique floue a été introduit par Lotfi Zadeh en 1965 [Zadeh (1965)], particulièrement pour prendre en charge autant les variables de type linguistique que les variables quantitatives, et de modéliser des systèmes complexes employant des données imprécises, incomplètes et vagues, ou bien lorsque le système utilise des règles d'inférence qui agissent d'une manière très générale avec des catégories floues [Chen and Pham (2000)].

Dans la théorie des ensembles classiques, la *fonction d'appartenance* signifie qu'un élément est un membre ou non, et indique si un élément appartient ou non à un ensemble bien

défini. Cette fonction peut être simplement représentée par des valeurs booléennes « vrai » ou « faux » dont les limites de l'ensemble sont claires et précises pour permettre la distinction entre les entités.

Les ensembles conventionnels ne traitent pas les entités à appartenance partielle, par contre la théorie des ensembles flous permet de représenter mathématiquement les entités à appartenance partielles. La fonction d'appartenance prend ses valeurs dans l'intervalle  $[0,1]$ , ce qui permet de définir des valeurs intermédiaires entre les évaluations booléennes conventionnelles.



**Figure 2.5** – Structure générale d'un système d'inférence floue  
(Source : [Paris et al. (2008)])

Le processus décisionnel dans un système flou s'appuie sur un système d'inférence floue, qui comprend les fonctions suivantes: La fuzzification (modification logique floue) permettant de représenter toute connaissance ou variable linguistique relative à la quantité ou au sentiment, telle que large, grand, heureux, affamé, etc., à l'aide d'un modèle mathématique transformant des entrées nettes en entrées floues (degrés de correspondance des valeurs linguistiques); alors que la défuzzification (clarification) convertit les valeurs floues en valeurs nettes, produisant des résultats quantifiables. Quant au moteur d'inférence floue, il effectue des opérations d'inférence en se basant sur les règles floues (Figure 2.5).

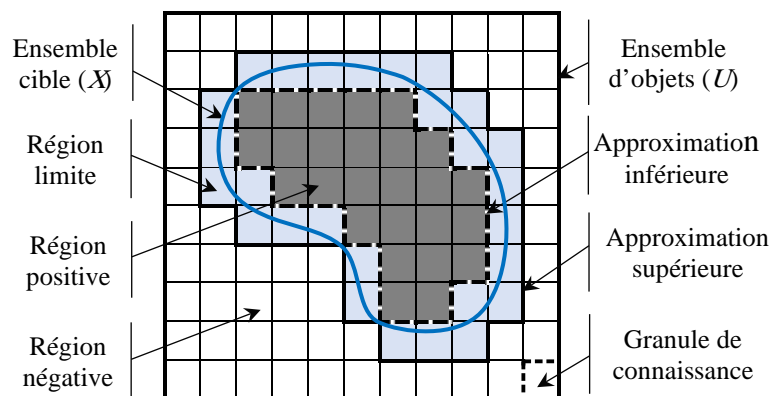
Dans les SIAD, les techniques basées sur la logique floue sont plus pratiques quand elles sont combinées avec d'autres techniques issues de l'intelligence artificielle.

### ***2.3.5. Les systèmes basés sur les ensembles approximatifs (rough sets)***

Les systèmes d'aide à la décision classiques fonctionnent parfaitement avec des informations précises et exhaustives, mais lorsqu'il est question de manipuler des informations imprécises, incomplètes ou incertaines, il est difficile de traiter ces données et d'en tirer des conclusions. Pour traiter des problèmes du monde réel utilisant des connaissances inexacts, incertaines ou vagues, les théories des ensembles approximatifs et des ensembles flous se sont montrées comme des outils puissants.

En tant qu'approche mathématique, la théorie des ensembles approximatifs a été introduite par Zdzislaw Pawlak en 1982 pour couvrir les connaissances imparfaites.

Au lieu d'exprimer le flou par une composition floue, la théorie des ensembles approximatifs utilise la notion de limite de région d'un ensemble, qui est définie par la différence entre les éléments qui appartiennent probablement et ceux qui appartiennent certainement (c'est-à-dire la différence entre l'approximation supérieure et l'approximation inférieure) (Figure 2.6).



**Figure 2.6** – Ensembles approximatifs (Rough set)

Le manque de connaissances dans un système d'information rend difficile la distinction de certains objets, d'où le concept de la théorie des ensembles approximatifs qui a introduit la relation d'indiscernabilité, qui définit la relation entre les objets concernés dont les valeurs sont perçues comme identiques par rapport à un sous-ensemble d'attributs.

Par conséquent, cette relation d'équivalence produit une topologie représentant les opérations d'intérieur et de fermeture, également appelées approximations respectivement inférieure et supérieure (Figure 2.6).

En respectant la relation d'indiscernabilité, la région limite est un ensemble d'objets formé de la différence entre l'ensemble d'approximation supérieur et l'ensemble d'approximation inférieur. La région limite vide signifie que l'ensemble considéré est net, sinon l'ensemble est approximatif, autrement dit les connaissances véhiculées ne sont pas assez suffisantes pour définir parfaitement cet ensemble [Raza and Qamar (2017), Liu and Zhu (2008), Pawlak (1982), Pawlak and Skowron (2007), Greco et al. (2001), Rissino and Lambert-Torres (2009), Walczak and Massart (1999)].

L'approche des ensembles approximatifs utilise des algorithmes et des outils efficaces permettant de trouver des modèles (patterns) cachés dans les données, de réduire les données sources à des ensembles minimaux, et de générer automatiquement des règles de décision permettant une interprétation facile des résultats obtenus. D'autant plus cette approche ne nécessite pas d'informations supplémentaires relatives aux données comme

les probabilités dans les statistiques et le degré d'appartenance dans la théorie des ensembles flous.

La théorie des ensembles approximatifs a été largement utilisée dans la sélection des caractéristiques des données, en ne conservant que celles qui sont représentatives. Ses algorithmes sont adaptés au traitement parallèle et trouvent leurs applications dans différents domaines de l'intelligence artificielle et dans les systèmes d'aide à la décision notamment la reconnaissance et classification des formes, le diagnostic médical, l'analyse acoustique, l'analyse de sécurité et les systèmes de contrôle intelligents, etc.

### 2.3.6. Les systèmes basés sur l'informatique évolutionnaire

Inspirée de la théorie de l'évolution et de la biogénétique, l'informatique évolutionnaire vise les problèmes d'optimisation particulièrement les problèmes combinatoires, les problèmes de la dynamique non-linéaire et de la recherche opérationnelle.

En faisant appel à divers algorithmes d'optimisation tels que l'algorithme génétique (AG) (Figure 2.7), la programmation génétique (PG), la programmation évolutionnaire (PE), ainsi que les variantes de l'intelligence d'essaim appelée aussi intelligence collective (*swarm intelligence*) [Krause et al. (2013)] par exemple, l'optimisation en essaim de particules, et optimisation par les colonies de fourmis (OCF). Cette théorie permet d'achever rapidement le processus d'optimisation en retournant des solutions de bonne qualité. La combinaison de ces algorithmes avec des systèmes d'aide à la décision produit des systèmes d'aide à la décision intelligents utiles pour pallier une multitude de problèmes d'optimisation du monde réel.

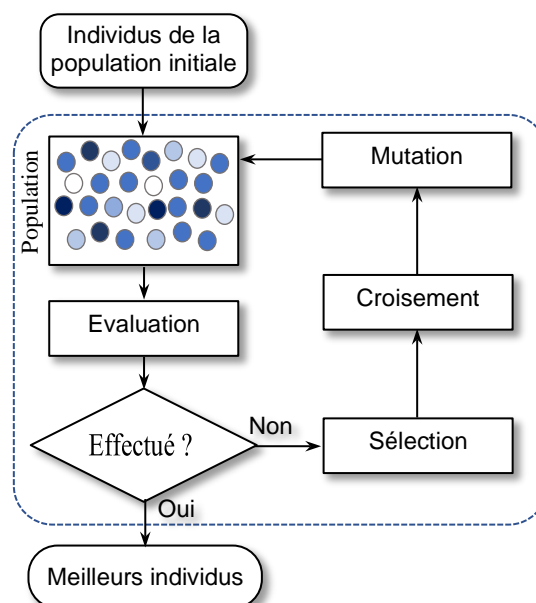


Figure 2.7 – Diagramme général de l'algorithme génétique

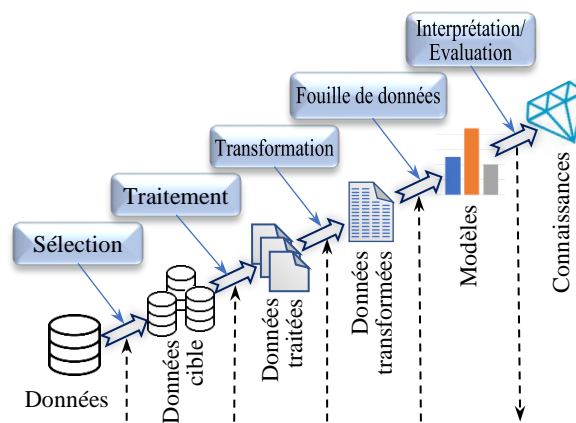
Dans sa version simple, l'algorithme génétique se distingue par les trois opérateurs génétiques suivants : la *sélection*, le *croisement* et la *mutation*. L'AG fonctionne de manière itérative pour faire évoluer les individus d'une population sur une succession de générations, dans le but de produire une progéniture d'individus pour chaque génération, puis l'utiliser pour former la population suivante [Kramer (2017), Pétrowski and Ben-Hamida (2017)].

### 2.3.7. La fouille des données et la fouille des traitements

Les données numériques ont connu une croissance exponentielle dans tous les domaines où elles intègrent une quantité importante d'informations et de connaissances cachées. C'est pourquoi de nombreuses théories et outils informatiques sont employés pour explorer et exploiter ces informations, et particulièrement dans la découverte de connaissances dans une base de données (*knowledge discovery in database KDD*) [Fayyad et al. (1996)], qui vise à trouver des modèles et des relations significatifs à partir des jeux de données et cela à travers les étapes suivantes (sélection de données, traitement de données, transformation de données, fouille de données, et interprétation et évaluation des données) (Figure 2.8).

En tant qu'étape d'analyse de KDD, la fouille de données est un processus vivant d'extraction d'informations utiles et compréhensibles (connaissances) à partir des ensembles de données volumineux. Il s'agit donc d'un domaine interdisciplinaire qui associe les systèmes de gestion de bases de données, les modèles statistiques, les algorithmes mathématiques et les méthodes d'apprentissage automatique (*machine learning ML*).

Grâce à ses méthodes comme la généralisation, la caractérisation, la classification, la segmentation, l'association, la correspondance de modèles et la visualisation de données, La fouille de données sert d'outil d'aide à la décision qu'on pourrait appliquer dans divers domaines, notamment l'industrie, la finance, le commerce électronique, l'ingénierie et la biomédecine.



**Figure 2.8** – Processus de la découverte de connaissances dans une base de données (source : [Fayyad et al. (1996)])



Tandis que la fouille de données est une approche centrée sur les données, utilisée pour extraire des connaissances inconnues au préalable à partir des données brutes, la fouille des processus consiste à extraire et à améliorer les connaissances relatives aux processus à partir des journaux d'événements. Il s'agit d'un processus d'une approche basé-modèles c'est-à-dire découvrir des modèles de processus en observant des événements de données. La fouille des processus [De Leoni et al. (2016)] a connu de nombreuses applications dans différents domaines (commerce, industrie, santé, services) partageant un comportement dynamique lié aux modèles de processus.

### 2.3.8. L'apprentissage machine et l'apprentissage profond

L'apprentissage automatique est considéré comme un axe de recherche très actif de l'intelligence artificielle. Inventé par Arthur Samuel en 1959 [Samuel (1967a), Samuel (1967b)] pour regrouper des techniques basées sur l'expérience acquise en étudiant des données ou en appliquant des règles connues permettant de catégoriser les objets, de prédire des résultats, d'identifier les modèles ou de détecter les comportements inattendus. L'apprentissage automatique permet aux ordinateurs d'apprendre automatiquement et de s'améliorer progressivement pour prendre de meilleures décisions en imitant les actes humains.

Sans compter sur une programmation explicite ou sur une programmation basée règles, les algorithmes d'apprentissage automatique visent à prédire les résultats à l'aide des données et des exemples concrets.

En fonction de leur objectifs, les algorithmes d'apprentissage automatique sont classés selon les types suivants : l'apprentissage supervisé, l'apprentissage non supervisé, l'apprentissage semi-supervisé, l'apprentissage par renforcement (Figure 2.9).

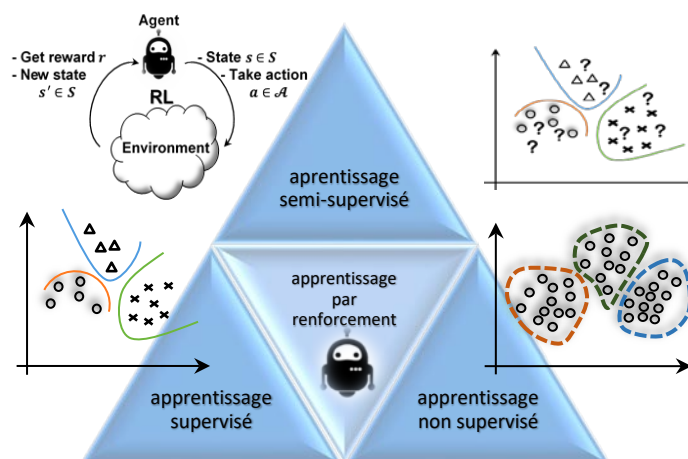


Figure 2.9 – Les types d'apprentissage machine

L'apprentissage automatique supervisé [Kotsiantis et al. (2007)] s'appuie sur des exemples étiquetés, afin de produire une fonction d'inférence capable de prédire une réponse précise pour les nouvelles entrées. Ses algorithmes comprennent des

classifications dont la cible est une valeur qualitative ainsi que des régressions dont la cible est une valeur numérique.

Ce type d'apprentissage automatique trouve son application dans la reconnaissance d'images, des textes, et de la parole, la détection des fraudes d'identité, les prévisions météorologiques, les prévisions de marché et les prévisions de croissance démographique, etc. Tandis que l'apprentissage automatique non supervisé [Ghahramani (2003), Sathya and Abraham (2013)] s'applique sur des exemples non étiquetés.

Le processus consiste à restructurer ces données en explorant leurs caractéristiques, puis en déduire des conclusions pour découvrir des modèles cachés fournissant des informations significatives. La segmentation et la réduction de la dimensionnalité sont les méthodes les plus connues d'apprentissage automatique non supervisé, elles s'appliquent surtout dans la détection d'anomalies ou d'intrusions, identification des éléments similaires, et l'analyse du marché.

Lorsqu'il est difficile de fournir des réponses à toutes les données en entrées, les algorithmes d'apprentissage automatique semi-supervisés [Zhu (2005), Huang et al. (2014)] utilisent à la fois des exemples étiquetés et des exemples non étiquetés. Dans la pratique, un échantillon plus large reflète mieux la distribution de probabilité des données, néanmoins l'étiquetage des données nécessite des ressources supplémentaires ; C'est pour cela, il suffit d'étiqueter une partie de l'ensemble des données puis les analyser afin d'obtenir un schéma général à appliquer pour extrapoler les données restantes.

En cas de manque de données d'apprentissage, l'apprentissage par renforcement [Sutton and Barto (2018)] intervient par le processus *essai et erreur*, basé sur les actions effectuées par l'agent afin d'obtenir des récompenses lors de l'interaction avec son environnement. Donc, l'agent s'entraîne à associer les situations aux actions de façon à maximiser le signal de récompense numérique, et par conséquent construire une stratégie permettant d'améliorer ses performances.

Au cours des dernières années, l'apprentissage par renforcement a résout un grand nombre de problèmes dans diverses applications, telles que la conduite autonome des voitures, les jeux d'intelligence (jeux échecs, TD-gammon, Go), la robotique... etc.

Un nouveau paradigme de l'apprentissage automatique dont le nom est l'apprentissage profond [LeCun et al. (2015), Goodfellow et al. (2016)], repose en particulier sur les réseaux de neurones étendus comportant un nombre considérable de couches permettant de traiter des données massives. Bien que l'apprentissage profond ait prouvé son succès de servir efficacement de nombreux domaines, il nécessite assez de temps et de ressources pour accomplir le processus d'apprentissage et construire des modèles précis.

Il existe plusieurs techniques dans le domaine d'aide à la décision, dont les plus répandues sont citées ci-dessus. Vu l'importance de l'optimisation et son application pour résoudre des problèmes combinatoires complexes et à grande échelle, nous allons nous concentrer

dans la section suivante sur les techniques d'optimisation combinatoire et leur utilité dans les SIADs.

## 2.4. L'optimisation combinatoire et les SIADs

Généralement, un processus d'aide à la décision comprend les étapes suivantes : identifier le problème, construire les préférences, évaluer les alternatives, puis déterminer les meilleures alternatives [Simon (1977), Keeney et al. (1993), Kleindorfer et al. (1993)].

En fonction de la taille du problème et du nombre d'objectifs, les alternatives peuvent se multiplier de façon exponentielle, ce qui nécessite l'emploi des modèles de programmation mathématique [Tzeng and Huang (2013)] permettant de choisir, parmi les solutions possibles (réalisables), les meilleures solutions qui maximisent/minimisent un ou plusieurs objectifs tout en respectant certaines contraintes, c'est ce qu'on appelle un problème d'optimisation. La programmation mathématique classique se concentre plutôt sur les problèmes d'optimisation combinatoire (en anglais, *Combinatorial optimization problem* COP) à un seul objectif que sur les problèmes d'optimisation multi-objectifs rencontrés souvent dans le monde réel. Ce qui privilège davantage la recherche dans le domaine d'optimisation et particulièrement quand il s'agit de problèmes d'optimisation multi-objectifs, complexes et à grande échelle.

### 2.4.1. Les problèmes classiques de l'optimisation combinatoire

La description d'un problème d'optimisation peut conduire à plusieurs formulations bien différentes qui mèneront toutes à des résolutions différentes du même problème. La recherche dans le domaine de l'optimisation a mis en évidence les différents types de problèmes d'optimisation à considérer comme étant des références ou encore des *benchmarks* dans la recherche opérationnelle et l'intelligence artificielle. Parmi ces problèmes d'optimisation les plus connus on peut citer : le plus court chemin (en anglais, *shortest street route problem*), le voyageur de commerce (en anglais, *traveling salesman problem* TSP), acheminement des véhicules (en anglais, *vehicle routing problem*), sélection d'équipe (en anglais, *team selection problem*), allocation des ressources (en anglais, *resource allocation problem*), planning d'emploi (en anglais, *job shop scheduling problem*), ainsi que le problème du sac-à-dos (en anglais *Knapsack problem* KP) et ses variantes. Ce dernier sera abordé en détails dans les sections qui suivent, comme étant un support convenable permettant de formuler notre problème de planification des projets urbains suivant le contexte de cette thèse.

Un problème d'optimisation combinatoire vise à rechercher en fonction des objectifs l'élément *max* ou *min* dans un ensemble fini. Etant donné une famille  $\mathcal{F}$  de sous-ensembles d'un ensemble fini  $E = \{e_1, \dots, e_n\}$  et un système de poids  $w = (w_{(e_1)}, \dots, w_{(e_n)})$  associé aux éléments de  $E$ . Admettons qu'il s'agisse de minimiser la fonction objective, le problème d'optimisation consiste alors à trouver un ensemble  $F \in \mathcal{F}$  de poids  $w(F) = \sum_{e \in F} w(e)$  minimum, c'est-à-dire :

$$\min \{w(F) | F \in \mathcal{F}\}$$

La famille  $\mathcal{F}$  représente donc les solutions du problème. Elle peut correspondre à un ensemble de très grande taille que l'on ne connaît que par des descriptions ou des propriétés théoriques qui ne permettent pas facilement son énumération, ainsi le problème devient complexe [Fouilhoux (2015)].

Un problème est dit très difficile (complexe), quand il est de type *NP*-difficile au sens propre du mot, ou bien quand il n'y a pas une implémentation d'algorithme polynomial exact pour résoudre ce problème<sup>6</sup>. En effet, la tâche de résoudre de manière exacte ce genre de problèmes intervient incessamment, et grâce à de nombreuses méthodes commensurables par leur degré de complexité y compris celles de la recherche exhaustive telles que la séparation-évaluation progressive (*branch and bound*), la programmation dynamique, etc.,

Afin de traiter des problèmes de grande dimension, ainsi que les problèmes complexes d'optimisation combinatoire plusieurs méthodes approximatives, heuristiques et méta-heuristiques sont utilisées comme :

- Séparation-évaluation progressive
- Les méthodes d'approximation constructive : règles de distribution de priorité y compris la logique floue, l'algorithme d'insertion de Werner and Winkler avec la recherche en faisceau (*beam search*).
- Les procédures de la recherche heuristique approximatives.
- Les méthodes de la recherche locale itérative comprenant en particulier, la méthode de vecteur de récession (*recession vector method*), la recherche d'équilibre global.

Les méthodes itératives de l'intelligence artificielle sont les suivantes : les algorithmes génétiques, la recherche locale génétique, satisfaction des contraintes y compris l'utilisation des réseaux de neurones ; l'optimisation par colonies de fourmis ; la recherche tabou ; les méthodes de problème d'espace et les heuristiques y compris les algorithmes gloutons ; les algorithmes de seuil : le recuit simulé (*simulated annealing*) [Zgurovsky and Pavlov (2018)].

#### ***2.4.2. L'optimisation combinatoire et le problème du sac-à-dos***

Chacun de ces types représente une famille de problèmes d'optimisation du monde réel. A titre d'exemple, le problème du sac-à-dos a connu récemment un grand succès [Laabadi et al. (2018), Hojny et al. (2019)] dans une multitude de publications de la recherche scientifique couvrant différents centres d'intérêt comme les algorithmes exacts [Furini et al. (2018)], les heuristiques [Haddar et al. (2016)] et les schémas d'approximation [Giudici et al. (2017)].

---

<sup>6</sup> Il n'est pas prouvé que le problème appartienne à la classe *NP*-difficile, un algorithme polynomial exact approprié n'existe pas, et les chercheurs n'ont pas pu fournir un tel algorithme.

Néanmoins, l'extension d'un problème d'optimisation vers des dimensions plus grandes qui relèvent de la pratique pourrait conduire à des changements importants. En fait, l'extension de problème du sac-à-dos en termes de nombre de sacs-à-dos, contraintes, et fonction objectif nécessite la modification de la structure du problème par rapport à l'ensemble des éléments (*item set*).

Il s'avère que la définition du problème du sac-à-dos est facile à comprendre même par des non experts, qui n'estimeront pas à la première rencontre, la présence d'un défi sérieux dans un tel sujet de recherche. D'où une attention particulière a été accordée à plusieurs variantes et extensions du problème du sac-à-dos et à l'étude de ses aspects stochastiques.

En effet, il convient de s'en servir dans cette thèse pour modéliser la planification des projets urbains, visant une meilleure adéquation des projets aux zones urbaines appropriées. Donc, il est utile de déterminer le nombre d'unités élémentaires de chaque type de projet formant un plan urbain, qui couvre un ensemble de zones urbaines, satisfaisant les besoins des différents acteurs, et respectant les contraintes limitant l'exploitation des ressources du territoire étudié.

En tenant compte des acteurs entrant en jeu, l'objectif est d'établir un plan urbain *optimal*, qui à la fois réponde aux exigences du niveau local et satisfait les directives et orientations du niveau global.

Dans la littérature, le problème du sac-à-dos couvre les variantes suivantes : la somme des sous-ensembles, le sac-à-dos borné, le sac-à-dos non borné, ainsi que les sacs-à-dos multidimensionnels, multiple et à choix multiple et le sac-à-dos quadratique [Kellerer et al. (2004)].

La description du problème de la planification urbaine faisant objet de notre étude, nous a permis d'opter pour la variante sac-à-dos borné (en anglais, *bounded knap sack problem* BKP), comme étant un outil d'appui pour la sélection rationnelle d'unités de projets. Chaque type de projet comprend un nombre limité d'unités identiques caractérisées par leurs préférences et leurs coûts.

Mathématiquement, les données du problème du sac-à-dos peuvent être exprimées comme suit :

Les objets sont numérotés par l'indice  $i$  variant de 1 à  $n$ . Les variables  $w_i$  et  $p_i$  représentent respectivement le poids et la valeur (ou le profit) de l'objet  $i$ . La capacité du sac sera notée par  $W$ . L'objectif est de remplir le sac-à-dos de façon à prendre le maximum d'objets de valeur sans dépasser la capacité du sac.

Généralement, on utilise un codage binaire, affectant la valeur 1 à chaque élément mis dans le sac, et la valeur 0 pour un élément non retenu. Soit le vecteur  $x = (x_1, x_2, \dots, x_n)$  décrivant l'état de remplissage du sac-à-dos, où  $x_i$  représente la valeur binaire associé à l' $i$ -ème élément. Cette variante est appelée sac-à-dos en variables binaires (0-1 KP).

La valeur totale des éléments mis dans le sac est calculée par la fonction :

$$z(x) = \sum_{i,x_i=1} p_i = \sum_{i=1}^n p_i \cdot x_i \quad (1)$$

De même la somme des poids des éléments choisis est donnée par la fonction :

$$w(x) = \sum_{i,x_i=1} w_i = \sum_{i=1}^n w_i \cdot x_i \quad (2)$$

Donc, le problème consiste à rechercher le vecteur d'éléments  $x$  *optimal*, qui maximise la *fonction objectif*  $z(x)$  sous la *contrainte* définie par la fonction  $w(x)$ .

Formellement le problème s'écrit par l'équation suivante :

$$\begin{cases} \max z(x) = \sum_{i=1}^n p_i \cdot x_i, & p_i \in \mathbb{N}^* \\ \text{sous contraintes :} \\ \sum_{i=1}^n w_i \cdot x_i \leq W, & W, w_i \in \mathbb{N}^* \\ x_i \in \{0, 1\} \forall i \in \{1, \dots, n\} \end{cases} \quad (3)$$

Pour éviter les cas triviaux, d'autres contraintes sont à prendre en compte :

- $w_i \leq W, \forall i \in \{1, \dots, n\}$  : aucun élément ne dépasse le poids du sac ;
- $\sum_{i=1}^n w_i > W$  : la capacité du sac ne supporte pas tous les objets simultanément.

### 2.4.3. Les variantes du problème de sac-à-dos

En fait, le sac-à-dos en variables binaires (0-1 KP) est le plus simple des problèmes non triviaux d'optimisation linéaire en variables binaires : une seule contrainte et uniquement des coefficients entiers positifs. Il s'agit d'une variante parmi d'autres, dans ce qui suit nous allons présenter quelques variantes du problème du sac à dos qui diffèrent par le domaine des variables, les valeurs des objets, le nombre des sacs-à-dos, etc. Ici, on s'intéresse particulièrement aux sacs à dos en variables entières (discrètes).

\* **sac-à-dos multidimensionnel** (*multidimensional knapsack problem* d-KP) :

Comme son nom l'indique, ce type de problème noté par (d-KP) possède des contraintes de capacité avec  $d > 0$  dimensions, il peut être considéré comme un programme linéaire en nombres entiers (ILP) à variables binaires. Les contraintes sont formulées par :

$$\begin{cases} \max z(x) = \sum_{i=1}^n p_i \cdot x_i, & p_i \in \mathbb{N}^* \\ \text{s.c. } \sum_{i=1}^n w_i^j \cdot x_i \leq W_j, & W_j, w_i \in \mathbb{N}^d, j \in \{1, \dots, d\} \\ x_i \in \{0, 1\} \forall i \in \{1, \dots, n\} \end{cases} \quad (4)$$

\* **Sac-à-dos multi-objectifs** (en anglais, *multi-objective knapsack problem* MOKP) : consiste à optimiser  $m > 1$  fonctions objectifs à partir des valeurs des objets. Cette variante est formulée par :

$$\begin{cases} \max z^j = \sum_{i=1}^n x_i \cdot p_i^j & j \in \{1, \dots, m\} \\ \text{s.c. } \sum_{i=1}^n x_i \cdot w_i \leq W \\ x_i \in \{0, 1\}, \forall i \in \{1, \dots, n\} \end{cases} \quad (5)$$

\* **Sac à dos quadratique** (en anglais, *quadratic knapsack problem* QKP) : il s'agit de prendre en considération la relation de préférence entre les objets pris. Le gain supplémentaire  $g_{i,j}$  apporté par le choix simultané de deux objets rentre dans la fonction objectif :

$$\max z(x) = \sum_{i=1}^n x_i \cdot p_i + \sum_{i=1}^n \sum_{j=1}^n x_i \cdot x_j \cdot g_{i,j} \quad (6)$$

\* **Sac à dos à choix multiple** (en anglais *multiple choice knapsack problem* MCKP) : dans cette variante, on ne peut choisir qu'un seul objet représentant un groupe ou une classe d'objets, de manière à maximiser le profit. D'où la formulation du problème est :

$$\begin{cases} \max z(x) = \sum_{j=1}^k \sum_{i \in N_j} x_i^j \cdot p_i^j \\ \text{s. c. } \sum_{j=1}^k \sum_{i \in N_j} x_i^j \cdot w_i^j \leq W \\ \sum_{i \in N_j} x_i^j \leq 1, \forall j \in \{1, \dots, k\} \end{cases} \quad (7)$$

$k$  nombre de classe d'objets,  $N_j$  l'ensemble des indices des objets appartenant à la classe  $j$ . Il est à noter qu'un objet n'appartient qu'à une unique classe

\* **Sac à dos multiple** (en anglais, *multiple knapsack problem* MKP) : Cette variante consiste à généraliser la version standard du problème à partir d'un simple sac-à-dos à  $m$  sac-à-dos de capacités différentes.

Sachant que la valeur d'un objet dépend du sac dans lequel il est placé, l'objectif est d'affecter chaque objet au plus à un sac-à-dos, de telle sorte qu'aucune contrainte de capacité ne soit dépassée et que le total des poids de tous les objets mis dans les sacs soit maximisé. La formulation devient :

$$\begin{cases} \max z(x) = \sum_{j=1}^m \sum_{i=1}^n x_i^j \cdot p_i^j \\ \text{s. c. : } \sum_{i=1}^n x_i^j \cdot w_i \leq W^j \quad \forall j \in \{1, \dots, m\} \\ \sum_{i=1}^n x_i^j \leq 1 \quad \forall i \in \{1, \dots, n\} \end{cases} \quad (8)$$

\* **Sac-à-dos borné** (en anglais, *bounded knapsack problem* BKP) : la spécificité est de considérer le nombre de copies identiques  $u_i$  de chaque objet  $i$ . Si ce nombre de copies est limité on parle donc de sac-à-dos borné, sinon on a un sac à dos non borné (en anglais, *unbounded knapsack problem* UKP). Le BKP consiste donc à trouver le nombre de copies  $x_i$  à prendre pour chaque objet. Il est formulé par l'équation suivante :

$$\begin{cases} \max z(x) = \sum_{i=1}^n p_i \cdot x_i, \quad p_i \in \mathbb{N}^* \\ \text{s. c. } \sum_{i=1}^n w_i \cdot x_i \leq W, \quad W, w_i \in \mathbb{N}^* \\ 0 \leq x_i \leq u_i, \quad x_i \in \mathbb{N}^* \quad \forall i \in \{1, \dots, n\} \end{cases} \quad (9)$$

La sélection d'objets sous une contrainte de capacité dont le nombre de copies est limité pour chaque type d'objet est un modèle présent dans plusieurs problèmes de décisions pratiques [Kellerer et al. (2004)].

#### 2.4.4. La résolution des problèmes d'optimisation combinatoire

Pour résoudre un problème d'optimisation combinatoire, on fait appel soit aux méthodes *exactes* soit aux méthodes *approchées*. Certainement les méthodes exactes sont bien fondées mathématiquement, et en particulier en termes de convergence et d'optimalité globale. Le principe essentiel consiste généralement à énumérer exhaustivement l'ensemble des solutions de l'espace de recherche, où au moins s'assurer de ne pas écarter aucune solution

supposée être meilleure que la solution optimale trouvée. Elles permettent également de fournir des résultats exacts en un temps raisonnable pour des problèmes classiques tels que le *voyageur de commerce* et le *sac-à dos*.

Parmi ces méthodes on peut citer la programmation linéaire, quadratique ou dynamique, la méthode de séparation-évaluation progressive (en anglais, *branch and bound*), et la méthode du gradient.

Quant aux méthodes de résolution *approchées* comprenant en plus des *heuristiques*, les *méta-heuristiques*, apparues à la fin des années 80, elles ont prouvé leur performance de fournir des solutions de bonne qualité (*pseudo-optimales*) aux problèmes combinatoires avec un minimum de ressources<sup>7</sup>, d'autant plus, elles peuvent s'adapter facilement aux changements de la structure du problème comme l'ajout ou la suppression de contraintes [Benyamina (2013)]. Pour répondre aux exigences et particulièrement dans le domaine industriel, de nombreux outils d'aide à la décision intègrent des méta-heuristiques, dont les plus répandues sont les algorithmes génétiques, le recuit simulé, les réseaux de neurones, les algorithmes à estimation de distribution, l'optimisation par essaim de particules et l'optimisation par colonie de fourmis.

Le succès de méta-heuristiques revient à leurs propriétés fondamentales suivantes :

- Les méta-heuristiques visent à explorer l'espace de recherche pour trouver des solutions proches des solutions optimales.
- Les algorithmes méta-heuristiques peuvent aller de la recherche locale simple jusqu'au processus d'apprentissage complexes.
- Les méta-heuristiques peuvent être construites à base des heuristiques relatives au problème traité, qui sont guidées par une stratégie supérieure.
- Durant la recherche de l'optimalité, les méta-heuristiques exploitent l'expérience acquise pour mieux explorer l'espace de recherche.

Dans ce qui suit, nous allons présenter brièvement quelques méta-heuristiques notamment les méthodes de la recherche locale, ensuite nous abordons en détail les concepts fondamentaux de l'algorithme des colonies de fourmis.

Le principe de base de la recherche locale consiste à améliorer itérativement une solution initiale en explorant les solutions voisinant celle-ci dans l'espace de recherche. Parmi ces méthodes on trouve : la méthode de descente, le recuit simulé, et la recherche tabou.

**La méthode de descente :** afin d'explorer rapidement l'espace de recherche, cette méthode (dite aussi *basic local search* ou *hill climbing*) utilise une structure de voisinage comportant des règles de déplacement. A partir d'une solution  $s$ , Le processus s'itère et progresse à chaque fois, vers une solution voisine  $s'$  de meilleure qualité, jusqu'à ce que toutes les solutions de voisinage soient moins bonnes que la solution courante [Selman and Gomes (2006)]. La méthode de descente est la méthode de recherche locale la plus

---

<sup>7</sup> Contrairement aux méthodes exactes qui sont souvent trop coûteuses en termes du temps de calcul et d'espace mémoire, les méthodes méta-heuristiques réduisent davantage le temps de calcul.



élémentaire. Appelé aussi algorithme classique d'amélioration itérative, cet algorithme converge rapidement mais, ne conduit pas, en général, à l'optimum global, mais souvent à un optimum local.

**Le Recuit simulé :** L'idée est tirée du processus de refroidissement des métaux, on choisit une température initiale suffisamment élevée pour explorer librement l'espace de recherche, puis la température diminue petit à petit jusqu'à atteindre une valeur avoisinant le zéro (0). Inspiré de ce principe, *Kirkpatrick* a introduit cette méthode en 1983 dans le domaine d'optimisation [[Kirkpatrick et al. \(1983\)](#)], elle consiste à déplacer une solution  $s \in N(s)$  vers une solution  $s_0$  telle que  $F(s_0) > F(s)$ . Le recuit simulé a été appliqué avec succès à un grand nombre de problèmes d'optimisation NP-difficiles tels que, le problème d'affectation quadratique [[Misevicius \(2003\)](#), [Peng et al. \(1996\)](#)], et les problèmes d'ordonnancement de type job-shop [[Van Laarhoven et al. \(1992\)](#)]. Le recuit simulé a l'avantage de couvrir un espace de recherche plus grand et d'éviter la convergence prématurée vers un optimum local.

**La Recherche Tabou :** introduite par Glover [[Glover \(1986\)](#)], cette méthode ressemble beaucoup plus à la méthode de descente. Son fonctionnement considère un ensemble de solutions du voisinage de la solution courante, puis choisir la meilleure solution proche de cet ensemble, même si celle-ci est de qualité inférieure pour remplacer la solution courante. Il s'agit donc d'améliorer à chaque itération la valeur de la fonction objectif en croissant à chaque fois la meilleure solution voisine.

Afin d'éviter les cycles courts et restreindre le voisinage aux solutions les plus susceptibles de diriger l'exploration vers des régions inexplorées, cette méthode conserve à court terme dans une liste « liste tabou » les dernières solutions choisies pour ne plus les prendre en considération tant que celles-ci figurent encore dans la liste<sup>8</sup>.

Ayant moins de paramètres de réglage que la méthode précédente, cette méthode demeure simple à appliquer et fournit de bons résultats pour des problèmes difficiles. L'inconvénient de cette méthode est que si la solution atteinte est toute proche de l'optimum global, il y a possibilité de ne pouvoir pas échapper à l'optimum local atteint.

#### ***2.4.5. L'optimisation par les colonies de fourmis***

Durant les trois dernières décennies, de nombreuses recherches ont été menées sur les animaux et les insectes du monde naturel. C'est en observant de près leurs comportements pendant la recherche de nourriture, la division du travail et l'affectation des tâches entre les individus, qu'on a découvert une organisation très structurée et sans aucune supervision.

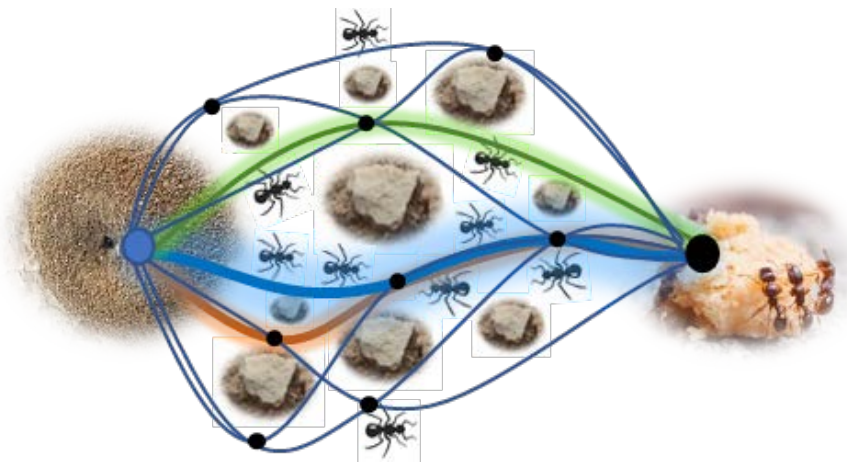
Ce comportement collectif de ces espèces constitue une source d'inspiration pour les scientifiques qui cherchent souvent à résoudre des problèmes complexes dans différents

---

<sup>8</sup> Pour une description complète de la méthode, le lecteur pourra utilement se reporter au papier de Glover et Laguna [[Glover and Laguna \(1998\)](#)].

domaines tels que la robotique et l'optimisation combinatoire (voyageur de commerce, coloration de graphes, élaboration des plannings, affectation quadratique...). L'intérêt a conduit au développement de plusieurs algorithmes basés sur l'intelligence collective ou l'intelligence d'essaim (en anglais, *swarm intelligence*), notamment les essaims de particules et les colonies de fourmis.

Dans cette thèse, nous allons nous concentrer sur l'utilisation de l'algorithme d'optimisation par les colonies de fourmis (*Ant Colony Optimization ACO*) » [Dorigo et al. (2000), Dorigo et al. (1991)]. Cette *méta-heuristique*<sup>9</sup> peut être adaptée et utilisée pour la modélisation et la résolution de la planification des projets urbains comme étant un problème d'optimisation combinatoire et/ou de satisfaction de contraintes.



**Figure 2.10** – Comportement des fourmis à la recherche de la nourriture  
(Source : [Khelifa and Laouar (2018)])

La popularité de ces algorithmes est due d'une part à la facilité de la mise en œuvre et d'autre part à son efficacité à résoudre des problèmes complexes [Deneubourg et al. (1991), Dorigo and Gambardella (1996), Deneubourg (1990), Colorni et al. (1992), Dorigo et al. (1996), Topin et al. (1999), Wang et al. (2020)].

Cette classe d'algorithmes est basée sur le comportement collectif des fourmis explorant des chemins, lorsque celles-ci sont à la recherche de nourriture, et communiquent entre elles au moyen d'une substance appelée *phéromone*, déposée par chaque fourmi tout au long du trajet. Sachant que la phéromone s'évapore continuellement, ceci influe sur le choix probabiliste des fourmis qui tracent leur chemin vers la source de la nourriture, en tenant compte de la quantité restante de phéromone.

<sup>9</sup> Le terme *méta-heuristique* a été introduit par Glover en 1986 [Glover (1986)], il est composé de deux mots grecs : *méta* qui signifie 'au-delà' c'est-à-dire, un passage à un niveau supérieur pour étudier ou manipuler des informations de niveau inférieur, et *heuriskein* qui veut dire 'trouver'.

Ces comportements simples traduits en interactions limitées à travers une coopération inconsciente, mènent à des comportements collectifs intelligents et à des modèles d'auto-organisation [Bonabeau et al. (1999)].

L'algorithme d'optimisation par colonies de fourmis (ACO) emploie des agents (*fourmis artificielles*) pour imiter les fourmis réelles. Depuis leur colonie, et sur leur chemin vers la source de la nourriture, les  $M$  fourmis déposent la phéromone sur les nœuds et/ou sur les arcs afin de marquer son itinéraire. Elles agissent en collaboration et communiquent ensemble dans une harmonie. A chaque itération de l'algorithme, chaque fourmi construit une solution complète, qui définit un chemin reliant les deux extrémités (la fourmilière et la source de la nourriture) sur le graphe (Figure 2.10).

Pour construire un chemin, la  $k$ -ième fourmi passe progressivement d'un état  $i$  à un état  $j$  par une probabilité  $P_{i,j}^k$  en ajoutant un composant à la solution partielle. Cette probabilité dépend de deux valeurs :

- La trace de phéromone  $\tau_{i,j}$  qui mesure la désirabilité apprise du mouvement.
- L'information heuristique  $\eta_{i,j}$  qui indique l'attractivité d'un mouvement.

$$P_{i,j}^k = \begin{cases} \frac{\tau_{i,j}^\alpha \eta_{i,j}^\beta}{\sum_{l \in \mathcal{N}_i} \tau_{i,l}^\alpha \eta_{i,l}^\beta}, & \forall j \in \mathcal{N}_i \\ 0, & \text{sinon} \end{cases} \quad (9)$$

Où  $\mathcal{N}_i$  est l'ensemble des nœuds voisins accessible à partir du nœud  $i$ . Les paramètres  $\alpha$  et  $\beta$  permettent d'ajuster respectivement les poids de la trace de la phéromone et la valeur de l'information heuristique.

La quantité de la phéromone diminue par le phénomène d'évaporation, donc la quantité restante est mise à jour par toutes les fourmis qui ont construit une solution durant la même itération. La quantité de la phéromone associée à l'arc joignant les nœuds  $i$  et  $j$  est calculée par :

$$\tau_{i,j} = (1 - \rho) \cdot \tau_{i,j} + \sum_{k=1}^M \Delta\tau_{i,j}^k \quad (10)$$

Où  $\rho$  est le taux d'évaporation et  $\Delta\tau_{i,j}^k$  est la quantité de la phéromone sur l'arc  $(i,j)$  déposée par la fourmi  $k$ , cette quantité est comme suit :

$$\Delta\tau_{i,j}^k = \begin{cases} Q/L_k, & \text{Si l'arc } (i,j) \in S_k \\ 0, & \text{Cas contraire} \end{cases} \quad (11)$$

Où, le constant positif  $Q$  est un paramètre pour normaliser les valeurs de la quantité de la phéromone,  $L_k$  est la longueur du chemin formant la solution  $S_k$  construite par la fourmi  $k$ .

L'information heuristique  $\eta_{i,j}$  représente à quel point, il est avantageux de prendre un chemin, c'est l'inverse de la distance  $d_{i,j}$  entre deux nœuds  $i$  et  $j$  :

$$\eta_{i,j} = \frac{1}{d_{i,j}} \quad (12)$$

Cette formulation représente l'algorithme basique d'optimisation par colonies de fourmis, appelé *Ant system* (AS), décrit comme suit (Algorithme 1) :

---

**Algorithme 1:** Basic ACO algorithm (AS)

---

```

Result: the best solution  $S^*$ 
Function ConstructAntSolutions ( $\tau$ )
begin
   $solution \leftarrow \emptyset$ 
  while solutionNotCompleted do
     $component \leftarrow \text{ConstructPolicy}(\tau, \eta)$ ;
     $solution \leftarrow solution \cup \{ component \}$ 
  end while
  return  $solution$ 
end
begin
  SetParameters ( $M, \alpha, \beta, \rho$ )
  InitializePheromoneTrails ( $\tau$ )
   $S^* \leftarrow \emptyset$ 
  while terminationConditionNotSatisfied do
     $S \leftarrow \emptyset$ 
    for each ant do
       $s \leftarrow \text{ConstructAntSolutions}(\tau)$ 
      ApplyLocalSearch ( $s$ ) #optional
       $S \leftarrow S \cup \{ s \}$ 
    end for
     $S^* \leftarrow \text{FindBest}(S \cup S^*)$ 
    Evaporation ( $\tau, \rho$ )
    UpdatePheromone ( $\tau, \text{select}(S \cup S^*)$ )
  end while
  return  $S^*$ 
end.

```

---

Au début, il faut initialiser les paramètres  $M, \alpha, \beta, \rho$  et l'information de la phéromone  $\tau$  puis répéter afin de construire la meilleure solution  $S^*$  dans un temps de calcul limité. A chaque itération, chacune des fourmis construit progressivement des solutions faisables en se déplaçant par probabilité d'un état à un autre. Quand une solution  $s$  est trouvée, elle pourrait être améliorée par une recherche locale et ajoutée à l'ensemble  $S$  des solutions. La meilleure solution  $S^*$  est mise à jour en cas d'une solution est meilleure que celle-ci. Avant la fin de la boucle, les quantités de la phéromone diminuent par l'effet d'évaporation, et suivant l'ensemble de solutions construites, ces quantités sont par conséquent mises à jour.

Depuis son apparition, plusieurs variantes de l'algorithme ACO ont été développées par exemple : *Elitist Ant System* (EAS), *Ant-Q*, *Ant Colony System* (ACS), *MAX-MIN Ant System* (MMAS), *Rank-based AS*, *ANTS*, *Best-Worst AS*, *Hyper-cube AS*, *Population-based ACO*, *Beam ACO*. La plupart de ces algorithmes s'adressent aux problèmes d'optimisation mono-objectif pour un seul critère [Yaseen and Al-Slamy (2008)]. Cependant les problèmes du monde réel nécessitent une optimisation simultanée de plusieurs critères

qui sont souvent contradictoires. D'où l'emploi des techniques d'optimisation multi-objectifs (en anglais, *multi-objective optimization* MOO).

Malgré sa tendance expérimentale, l'analyse théorique difficile et la distribution de probabilité change au cours des itérations, les algorithmes basés ACO ont montré des potentiels pour la découverte rapide des solutions de bonne qualité, et même si le temps d'optimisation est incertain la convergence est garantie. En plus de l'implémentation facile, ils s'adaptent à d'éventuels changements de problème seulement par la reformulation des distances relatives (les informations phéromonique et heuristique) [Selvi and Umarani (2010)]. De plus, son parallélisme intrinsèque donne un appui pour intégrer cet algorithme dans des modèles d'ingénierie complexes, et spécialement dans les systèmes multidisciplinaires, distribués et à grande échelle.

### 2.5. Synthèse des approches d'optimisation et travaux connexes

Dans le domaine d'aide à la décision, l'optimisation constitue un volet important pour résoudre les différents problèmes du monde réel que ce soient simples ou complexes. La complexité dépend principalement du degré de difficulté lié à plusieurs facteurs tels que le type et le domaine des données appropriées, la linéarité, la convexité, la dérivabilité, le nombre d'états, les optima locaux et le temps qu'il faut pour résoudre ce problème.

Cela dépend aussi de la dimension du problème (nombre de variables de décision, contraintes, multiplicité d'objectifs ...). En général, les problèmes faciles sont résolus en temps polynomial selon la taille du problème, tandis que les problèmes difficiles ne peuvent pas être résolus de manière exacte en un temps proportionnel par rapport à sa taille polynomiale. Une instance à grande échelle d'un problème polynomial pourrait être résolue dans un délai raisonnable, sinon elle sera insoluble.

Etant des approches principales d'optimisation, les méthodes exactes, heuristiques et d'approximation [Abdmouleh et al. (2017)] se sont montrées utiles. Les méthodes exactes telles que la méthode séparation-évaluation progressive et la programmation dynamique, la programmation linéaire en nombres entiers mixtes (*mixed integer linear programming* MILP) permettent de résoudre le problème d'optimisation combinatoire en énumérant simplement toutes les solutions possibles donnant la meilleure valeur de la fonction objective.

Malgré la spécificité de chaque méthode pour réduire l'espace de la recherche, ces méthodes restent limitées pour résoudre les problèmes du monde réel de grande dimension. Les méthodes d'approximation sont largement utilisées particulièrement pour résoudre les problèmes d'optimisation combinatoire très difficiles [Festa (2014)] qui génèrent des solutions pseudo-optimales d'une qualité éprouvée. Elles comprennent également les algorithmes gourmands (*greedy*), les algorithmes séquentiels, les algorithmes de la recherche locale, les algorithmes de la programmation linéaire, les algorithmes basés sur la relaxation ainsi que les algorithmes de la recherche aléatoire.

Lorsqu'il est difficile d'approximer, et particulièrement dans le cas de problème d'optimisation à grande échelle, les méthodes heuristiques et méta-heuristiques sont appliquées pour apporter des solutions pseudo-optimales dont la qualité ne peut être vérifiée que par l'expérimentation. Ici, les algorithmes génétiques sont les plus prometteurs, le recuit simulé, la recherche tabou, l'optimisation des essaims de particules, l'optimisation des colonies de fourmis, la recherche de variables de voisinage (*variable neighborhood search*), les procédures de la recherche gourmande adaptative aléatoire (*greedy randomized adaptive search procedures* GRASP), la recherche par dispersion, et la recherche d'harmonie [Maniezzo and Carbonaro (2002)].

Au cours des dernières années, les techniques d'optimisation évoluent fortement pour devenir le centre d'application dans de nombreux domaines. Quelles que soient les méthodes de recherche conventionnelles ou intelligentes, l'objectif est de résoudre avec plus de précision et d'efficacité, les problèmes d'optimisation non linéaires, multi-objectifs, à plusieurs niveaux, et à grande échelle. Par exemple, Cheng He et al., [He et al. (2019)] proposent un framework permettant d'accélérer l'efficacité de calcul de l'algorithme évolutionnaire pour des problèmes d'optimisation multi-objectifs à grande échelle, en le reformulant en un problème d'optimisation mono-objectif à faible dimension.

Les travaux de Xingjuan Cai et al., [Cai et al. (2019)] abordent aussi le problème d'optimisation à grande échelle avec une complexité croissante, en appliquant des mécanismes de sélection probabilistes sur de nombreuses variantes d'algorithme de chauve-souris (*bat algorithm* BA) en vue de garder celles qui ont de meilleures performances.

Ye Tian et al., [Tian et al. (2019)] proposent une recherche efficace basée sur un optimiseur d'essaim compétitif (*competitive swarm optimizer* CSO) pour résoudre des MOP à grande échelle. Il adopte une nouvelle stratégie de mise à jour des particules qui consiste en deux étapes pour mettre à jour de la position, ce qui peut améliorer grandement l'efficacité de la recherche.

Dans les travaux de Jiang Li et al., [Li et al. (2019)] ils proposent de nouvelles stratégies de mise à jour pour améliorer la qualité des solutions de l'algorithme d'optimisation des troupeaux d'éléphants (*elephant herd optimization* EHO) pour les problèmes d'optimisation à grande échelle, en exploitant les informations utiles des individus précédents dans le processus de la mise à jour ultérieure.

De même, Yin Zhang et al., [Zhang et al. (2020)] proposent un nouvel algorithme évolutionnaire multi-objectif basé sur la décomposition (*multi-objective evolutionary algorithm based on decomposition* MOEA/D) pour des problèmes d'optimisation à grande échelle, appelé MOEA/D-IFM, où les modèles d'information retournés (*Information Feedback Models* IFM) représentent une nouvelle stratégie qui peut incorporer des informations de l'itération précédente dans le processus de mise à jour. Tandis que, Justin

S. Gray et al., [Gray et al. (2019)] poursuivent leurs efforts pour améliorer l'outil d'optimisation de conception multidisciplinaire (*multidisciplinary design optimization* MDO), destinés aux problèmes de conception intégrant des modèles numériques couplés de systèmes d'ingénierie complexes. L'outil *open source* OpenMDAO, utilise des algorithmes de calcul de type-Newton pour les dérivés et exploite la structure du problème par de nouvelles stratégies hiérarchiques pour atteindre un niveau d'efficacité de calcul élevé, même pour des problèmes d'optimisation à grande échelle.

## 2.6. Conclusion


L'évolution des systèmes d'aide à la décision combinés aux techniques issues du domaine de l'intelligence artificielle ont permis de développer des systèmes informatiques puissants et intelligents. Etant doté des capacités cognitives, les SIADs fournissent un accès facile aux informations pertinentes et emploie un raisonnement similaire à celui de l'homme pour la prise de décision.

L'optimisation constitue une grande partie des SIADs, et grâce à l'efficacité de ses techniques dans la résolution de problèmes combinatoires complexes et à grande échelle, elle est largement sollicitée dans plusieurs domaines notamment les sciences de la nature, les mathématiques, l'industrie, les finances, etc.

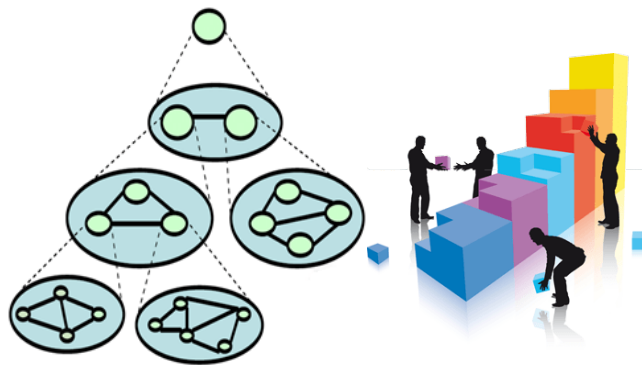
Notre domaine d'étude suscite l'emploi de ces techniques et d'en tirer profit pour planifier les projets urbains, notamment les techniques méta-heuristiques, telles que l'optimisation par colonies de fourmis, qui produisent des solutions pseudo-optimales comme étant un problème du sac-à-dos.

En effet, les algorithmes méta-heuristiques basés sur les populations de fourmis s'avèrent être des systèmes multi-agents dans le sens où les plans élaborés dans le cadre d'un problème de planification des projets urbains émergent de l'activité conjointe des fourmis qui peuvent être assimilées aux agents d'un système multi-agents.

Dans le chapitre suivant, nous allons se concentrer sur les concepts de base concernant l'approche holonique, offrant une organisation souple et flexible pour modéliser les projets urbains sous forme d'un système multi-agents holonique. Supportés par des algorithmes d'optimisation développés spécialement, les agents holoniques assurent une planification des projets urbains de manière cohérente et efficace.

 *Chapitre - 3*

# Une contribution basée SMAH pour la planification des projets urbains





## Chapitre - 3

# UNE CONTRIBUTION BASEE SMAH POUR LA PLANIFICATION DES PROJETS URBAINS

### 3.1. Introduction

Le changement urbain et l'extension remarquable des villes, font de la planification des projets urbains une priorité pour une gestion cohérente et efficace de l'ensemble des composantes du territoire urbain.

En effet, les systèmes d'aide à la décision pour la planification des projets urbains souvent reliés au pouvoir décisionnel centralisé, œuvrent continuellement à satisfaire les exigences informationnelles, organisationnelles, et fonctionnelles à travers une architecture distribuée.

Malgré l'intégrité organisationnelle de ces systèmes coopératifs, ils souffrent généralement d'un manque de cohérence globale [Adam and Kolski (1999)] entre ses composants, et particulièrement quand il s'agit des systèmes complexes à grande échelle. D'autant plus le pouvoir décisionnel centralisé ne répond pas parfaitement aux besoins réels exprimés par le niveau local.

Sans recourir à l'architecture centralisée ni à l'architecture purement distribuée, ce travail s'articule sur une architecture holonique combinant les deux architectures précédentes comme étant une solution au problème de la planification des projets urbains.

Nous nous sommes donc, dirigé vers un modèle organisationnel particulier basé sur l'approche holonique, offrant plus de cohérence, de flexibilité, et d'autonomie. Cette approche nous permettra de concevoir un système multi-agents holonique pour la planification des projets urbains assurant la coordination, la coopération, et l'échange des connaissances entre les agents dans une structure holarchique.

En fait, ce chapitre couvre deux volets, le premier est consacré aux concepts fondamentaux de l'approche holonique, tels que l'holon, l'holarchie, l'agent holonique, l'organisation holonique, ainsi que les systèmes multi-agents holoniques (SMAH) afin d'en tirer profit dans la modélisation du processus de la planification des projets urbains.

Le second volet comprend nos contributions, fondées sur l'application l'approche holonique dans la planification des projets urbains, en se servant d'une modélisation SMAH. Pour ce faire, les agents holoniques appliquent l'algorithme BKACS inspiré de l'algorithme basique de colonies de fourmis (*ant colony system* ACS), pour assurer l'optimisation locale des plans des projets urbains considérée comme étant un problème du sac-à-dos borné (*bounded knapsack problem* BKP), ainsi que l'algorithme H-MACO qui coordonne entre les holons des différents niveaux afin de réaliser une optimisation globale du système.

### 3.2. Le concept holon

A la fin des années soixante, le concept d'holon<sup>10</sup> a été introduit par le philosophe hongrois *Arthur Koestler* [Koestler (1967), Koestler (1969)], en étudiant les organismes vivants et les organisations sociales dans une tentative de réduire la dichotomie entre l'holisme<sup>11</sup> et réductionnisme<sup>12</sup>.

En partant d'une perspective d'ingénierie, l'approche holistique est la plus appropriée au paradigme des systèmes holoniques alors que les systèmes multi-agents (SMA) sont conçus selon la vision réductionniste. D'où l'émergence des systèmes holoniques qui se basent sur la notion d'holon plutôt que sur la notion d'agent.

Étant influencé par la théorie de l'évolution des systèmes simples vers des systèmes complexes de *Herbert Simon*, et à la suite des observations et analyse des hiérarchies du monde réel, *Koestler* a mis en évidence la relation *tout/partie* « *whole/part* » entre les entités. C'est-à-dire qu'il ne faut pas considérer seulement les concepts tels que les atomes, les molécules, les cellules, les individus, les systèmes, comme des unités autonomes ou indépendantes, mais aussi il faut admettre que chacune de ces unités est à la fois un tout – composé de parties similaires – et une partie d'un tout ou d'une organisation plus large [Mella (2009)]. En effet, ces unités représentent des *holons*.

En appliquant la relation *tout/partie* appelé aussi *contenant/contenu*, l'univers apparaît comme une hiérarchie d'holons, ce qui constitue une *holarchie* « *holarchy* » où les holons de chaque niveau hiérarchique subissent des changements structurels ou opérationnels des holons subordonnés, et à leurs tours produisent des changements de comportement des holons du niveau supérieur. L'interactivité se traduit par la communication bidirectionnelle entre tout et parties.

#### 3.2.1. Les caractéristiques des holons

Chaque holon doit avoir les caractéristiques suivantes : l'autonomie, l'autosuffisance, et la coopération [Koestler (1967)] ; un holon est autonome dans sa structure et son fonctionnement, lui offrant un état stable pour opérer indépendamment du contexte pour réaliser ses objectifs. Tandis qu'un holon est dit auto-suffisant s'il est capable de réagir et survivre même aux perturbations importantes de l'environnement sans se référer aux autorités supérieures. En ce qui concerne la coopération, un holon doit coexister et collaborer avec d'autres holons formant le même système pour atteindre les objectifs soulignés. Cependant, les holons restent sujet au contrôle d'un ou plusieurs holons supérieurs qui gèrent leur liberté d'action.

---

<sup>10</sup> L'acronyme holon est la concaténation du mot grec « *holos* » signifiant le tout et du suffixe « *on* » indiquant une partie.

<sup>11</sup> L'holisme signifie qu'un système quelconque (biologique, chimique, social, économique, linguistique, etc.) est un tout indivisible et ne peut pas être défini par ses composants considérés séparément.

<sup>12</sup> Le réductionnisme consiste à réduire un système complexe à ses composants plus simples et à considérer ces derniers comme plus fondamentaux que le système observé.

Il est donc capable à la fois de s'adapter aux circonstances contingentes sans avoir des permissions ou des instructions de la part des unités supérieures et de contrôler les unités subordonnées.

En plus de ces caractéristiques qui assurent la viabilité et la pérennité des systèmes sociaux, *Koestler* a mis d'autres règles et notions relatives concernant l'architecture holonique telles que la dualité coopération-autonomie et la communication [Adam et al. (1999)].

D'autres classifications et points de vue inspirés de la théorie de l'auto-transcendance et de la dynamique de l'univers ont généralisé le concept d'holon [Wilber (1995)], en évoquant l'organisation holarchique, considérant les caractéristiques fondamentales d'un holon :

- a) Auto-préservation : l'holon doit posséder des caractéristiques lui permettant de maintenir sa structure, son identité, et son organisation indépendamment des éléments qui le composent.
- b) Auto-adaptation : faisant partie d'une organisation plus large, l'holon doit être capable de s'adapter et d'établir des liens permettant de réagir avec ses super-holons.
- c) Auto-transcendance : l'holon possède ses propres propriétés qui diffèrent de celles des holons qui le composent. C'est créatif, quand de nouvelles propriétés émergent en conséquence de l'inclusion dans des holons supérieurs, d'où la naissance de nouvelles classes d'holons.
- d) Auto-dissolution : en suivant les mêmes lignes qui le forment, l'holon se divise verticalement ; le processus d'inclusion consécutive dans un sens ascendant se transforme en un processus de décomposition.

*Wilber* a proposé une classification des holons par nature qui pourraient être sensibles ou non-sensibles [Wilber (1995)]. Il en résulte quatre (04) types d'holons :

a) Holons sensibles :

- 1) Les holons individuels, ou holons au sens propre du mot : sont des entités possédant une intériorité localisée ou une conscience objective qui réalisent des activités autonomes ; à chaque niveau les holons seniors sont composés de holons juniors, et constituent des parties des holons du niveau supérieur.
- 2) Les holons sociaux : ils sont formés de groupes d'holons individuels ayant un modèle d'interaction stable. Ces holons n'ont pas une intériorité localisée ou une conscience objective, mais ils possèdent une existence autonome et une liaison de capacité d'agir.

b) Holons non sensibles :

- 1) Artéfacts ou systèmes physiques : ce sont des entités créées par des holons sensibles, par exemple les machines, les instruments, les langages, et les morceaux de musique...etc. Malgré le modèle organisé stable de leurs éléments constituants, ils n'ont pas de dimension intérieure, et ceci est dû au manque d'une des caractéristiques structurelles (auto-préservation, auto-adaptation, auto-transcendance).
- 2) Tas ou amas : des entités qui n'ont ni dimension intérieure ni modèle d'observation ou d'organisation stable. Si des holons sociaux créent des artéfacts sans en avoir l'intention, il devient alors difficile de distinguer les artéfacts des amas. Donc c'est l'intention et le niveau d'observation qui font considérer un tas comme un artéfact d'un holon social.

Selon *Wilber*, seulement les entités sensibles sont des holons typiques, et proprement dit les entités non-sensibles ne sont pas considérés comme des holons et ne forment pas des holarchies. En parallèle, *Andrew Smith* [Smith (2000)] a classé les holons en deux classes : les holons autonomes ou bien fondamentaux qui existent indépendamment des holons de l'ordre supérieur, et les holons intermédiaires ou sociaux qui n'existent seulement qu'avec des holons d'ordre supérieur.

Malgré la présence implicite du composant social dans chaque holon individuel, les holons sociaux ne représentent pas seulement une classe différente de celle des holons individuels, mais il s'avère qu'elle représente le plus haut niveau d'observation des holons individuels.

En généralisant le concept d'holon dans son sens formel, on peut le considérer comme une entité conceptuelle non-observable, qui agit comme étant un élément connectif ou intermédiaire des niveaux hiérarchiques de la *réalité* [Koestler (1967), Mella (2009)]. Ainsi, un holon est un point de référence donnant un sens hiérarchique à la réalité, appelée holarchie, qui est interconnectée à travers une inclusion dans des niveaux multiples.

Pourtant il est de la structure, l'holon n'est pas une structure, et ne correspond à aucune structure d'observation. Il est le centre des relations avec d'autres composants, subordonnés et composés, et structures supérieures (Figure 3.1).

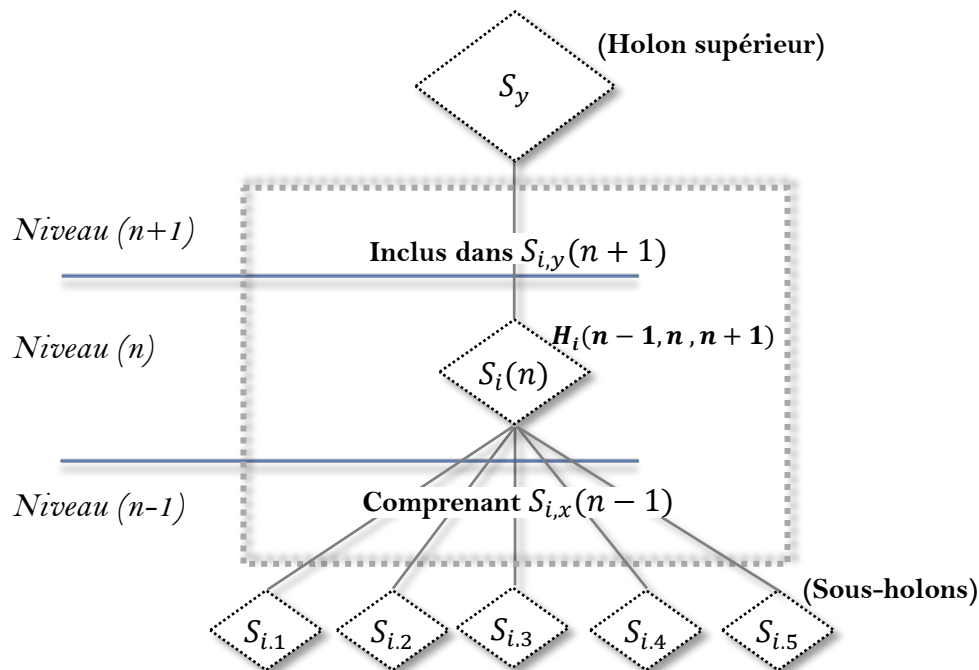


Figure 3.1 – L'holon dans une structure (Source : [Mella (2009)]).

Soit  $S_i(n)$  la  $i$ -ième structure ou système autonome, observable à l' $n$ -ième niveau. L'holon  $H_i(n-1, n, n+1)$  peut être vu comme étant une *description technique* de  $S_i(n)$  intégrée par la relation  $\supset$  (*comprenant*)  $S_{i,x}(n-1)$  indiquant toutes les structures  $x$  reliées à l' $i$ -ième structure du niveau  $(n-1)$  et la relation  $\subset$  (*inclus dans*)  $S_{i,y}(n+1)$  indiquant la

structure  $y$  du niveau  $(n + 1)$  à laquelle est reliée la  $i$ -ième structure  $S_{i,y}(n + 1)$  du niveau  $(n)$  ; Donc on note :

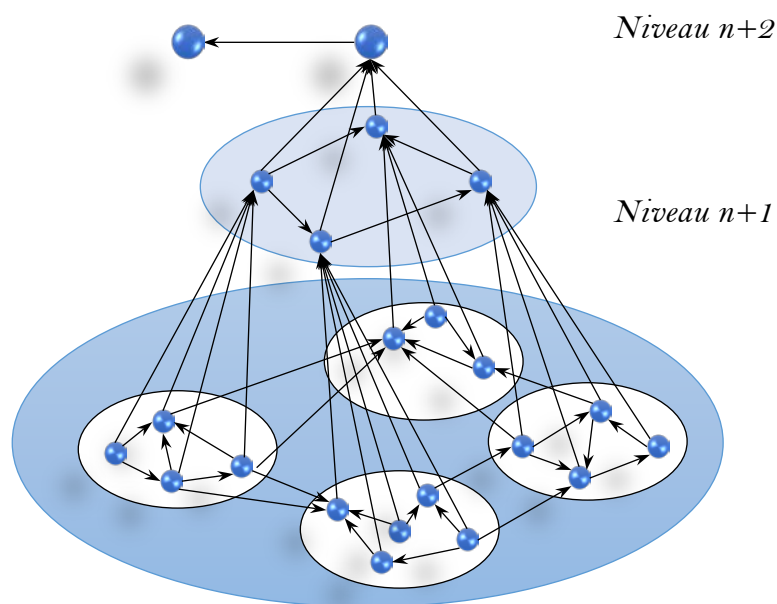
$$H_i(n - 1, n, n + 1) = \begin{cases} S_{i,x}(n - 1) \subset S_i(n) \\ S_{i,y}(n + 1) \supset S_i(n) \end{cases}$$

Finalement, l'holon est une entité structurante comprenant ses propres parties, aussi bien qu'une entité structurée faisant partie d'un autre holon, et qui n'est pas défini a priori. Ce n'est pas un objet physique, ni un individu, ni un système, mais il est une vision particulière de ces derniers. D'après ce point de vue holistique, toute chose peut être considérée comme un holon – un objet réel d'un système concret ou bien une entité conceptuelle d'un système de pensée – la réalité totale doit être observé d'une perspective holonique selon un arrangement vertical d'holons défini par des relations d'inclusion.

### 3.2.2. L'architecture holarchique

Etant donné que les holons d'un système sont décrits à la fois comme des ensembles auto-contenus en relation avec leurs parties subordonnées, et comme des parties dépendantes d'holons d'ordre supérieur. Ici, la notion d'inclusion signifie que chaque holon doit être inclus dans d'autres holons en accumulation progressive dans un ordre vertical typique, formant une organisation hiérarchique imbriquée appelée holarchie.

D'après, *Koestler* l'holarchie est définie comme étant une hiérarchie ouverte ordonnée d'holons autorégulés – *Self-regulating Open Hierarchic Order (SOHO)* – et organisés en coordination avec leur environnement local, sous forme d'ensembles supra-ordonnés de leurs composants, et en tant que parties subordonnées sous contrôle d'holons du niveau supérieur.



**Figure 3.2** – Organisation holarchique comme étant une structure multicouche (Selon [Ferber and Weiss (1999)])

Cette organisation d'holons peut être représentée comme une structure arborescente dont les branches s'élargissent vers le bas formant des niveaux d'hierarchies successifs, alors que les nœuds sont constitués d'agents holoniques (Figure 3.2).

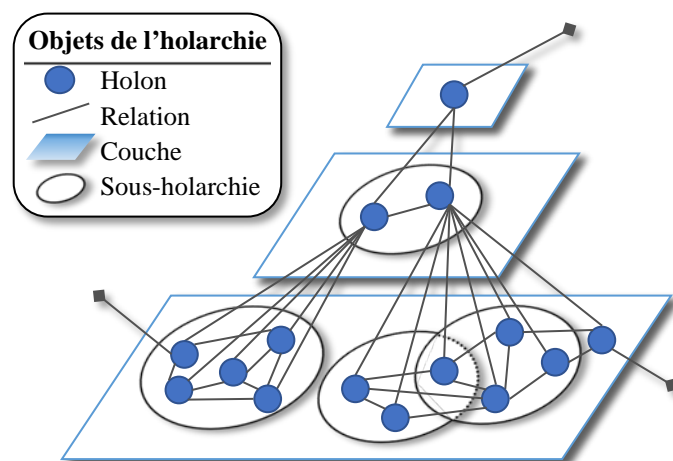
Chaque holon devient un holon tête « *head holon* » pour la branche sous-tendue et un holon membre pour la partie supérieure de la branche qui le forme.

Formellement les holarchies commencent par le niveau d'holons le plus bas (holons de base « *base holons* ») et se terminent aux niveaux les plus hauts (holon terminal ou final). Cette structure ouverte assure l'interconnexion avec l'environnement.

Grâce à la relation tout/partie, chaque holon est relié à la fois à un holon du niveau supérieur et aux holons du niveau inférieur, mais il n'a pas de liaison directe avec les holons du même niveau. Ces derniers ne se communiquent horizontalement qu'à travers les holons supérieurs.

En partant d'un holon primal, l'holarchie peut être produite par une décomposition successive descendante (les holons de base appartiennent au niveau ; l'holon final se trouve au niveau 0), comme elle pourrait être formée à partir d'un ensemble d'holons primaires dans une composition successive ascendante (les holons de base sont au niveau 0 tandis que l'holon final se trouve au niveau  $N$ ) [Mella (2009)].

En observant cette structure multicouches (multi-strates) du bas vers haut, chaque niveau hiérarchique comprend tous les holons de base, tandis que l'observation du haut vers le bas mène à une forme de segmentation du holon final (Figure 3.3).



**Figure 3.3** – Architecture d'une holarchie multicouches  
(Source [Calabrese et al. (2010)])

En effet, chaque niveau subordonné constitue une représentation minuscule des holons du niveau hiérarchiquement supérieur, où chaque holon tête est décomposé en holons subordonnés, bien que chaque niveau ne comprenne pas forcément tous les holons de base.

Notons le  $i$ -ième holon du  $n$ -ième niveau ( $n = 0, 1, 2, \dots, N - 1$ ) de la hiérarchie descendante, dont l'holon tête est  $H(0)$  (Figure 3.1). L'interaction d'un holon  $H_i(n)$  avec les autres éléments de l'holarchie doit obligatoirement respecter les règles suivantes [Mella (2009)] :

- A un niveau subordonné d'ordre  $(n + 1)$ , chaque holon entier « *whole* » auquel sont associés les holons composants  $H_{i,j}(n + 1)$ , où  $j$  dépend de  $i$ , compte les holons  $j$  faisant partie du  $i$ -ième holon ; donc  $H_{i,j}(n + 1)$  indique tous les holons du niveau  $(n + 1)$  inclus dans  $H_i(n)$ .
- A un niveau supérieur d'ordre  $(n - 1)$ , pour chaque  $H_i(n)$  du niveau  $n$ , il existe un et un seul holon entier  $H_{h,i}(n - 1)$  auquel  $H_i(n)$  est une partie.
- Seulement l'holon  $H_1(0)$  apparaît au niveau maximum (*niveau (0)*).
- Les holons  $H_{1,j}(1)$  qui font partie de  $H_1(0)$  apparaissent au niveau (1).
- Les holons  $H_{1,j,k}(2)$  qui font partie de  $H_{1,j}(1)$  apparaissent au niveau (2).
- Les holons  $H_{1,j,k,l}(3)$  qui font partie de  $H_{1,j,k}(2)$  apparaissent au niveau (3).
- Et ainsi de suite, de façon récursive, jusqu'au niveau minimum ( $N > 1$ ), où tous les holons de base inclus dans le niveau supérieur ( $N - 1$ ) sont contenus ; puis continuer récursivement vers le haut dans l'holarchie jusqu'au  $H_1(0)$ .
- Tous les holons de base sont reliés en quelque sorte à l'environnement ; même l'holon final est considéré connecté à l'environnement à partir duquel l'holarchie est isolée pour des raisons d'observation.

Dans chaque évènement de cette holarchie multi-strates s'applique le *principe de complétude*, c'est-à-dire tous les holons d'un certain niveau doivent être inclus dans ceux du niveau supérieur et comportent tous les holons du niveau inférieur.

Par conséquent, chaque niveau de l'holarchie comprend d'un part tous les holons de base et d'autre part représente une segmentation en parties du holon final.

Dans l'holarchie multi-niveaux, le *principe de congruence* s'applique dans le sens où tous les holons du même niveau doivent avoir des caractéristiques congruentes avec ceux des holons supérieurs, cela veut dire que tous les holons de base sont inclus dans l'holon final.

Comme dans la figure (Figure 3.4) un holon peut appartenir à n'importe quel niveau à l'exception du niveau le plus bas car il est indivisible.

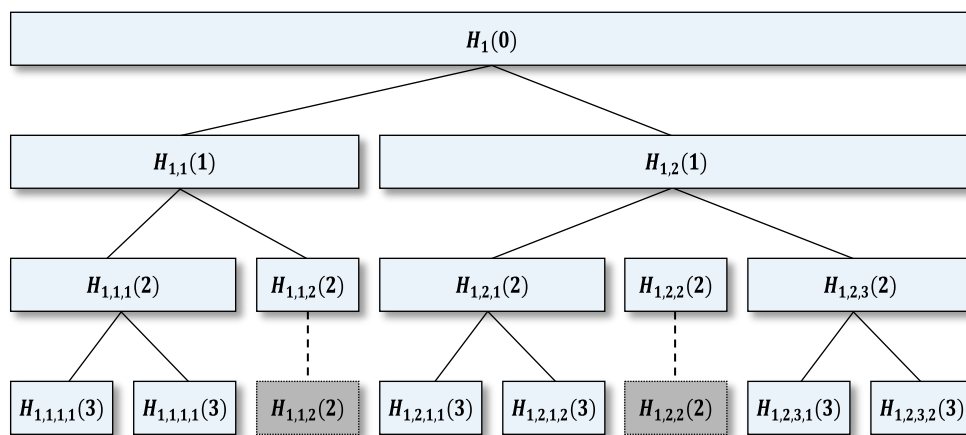
Afin de respecter le principe de complétude dans l'ordre des multi-strates, il est nécessaire d'introduire les holons virtuels (cellules en gris) dans l'arborescence [Mesarovic et al. (2000)], dont la fonction est de ramener au niveau le plus bas les holons non-décomposable, ou au niveau le plus haut les holons qui ne font pas partie directement de cette structure.

A chaque évènement, les propriétés des holons d'un niveau quelconque émergent en respectant celles des holons subordonnés.

Indépendamment de leur nature, la relation entre les holons du niveau ( $n$ ) et ceux du niveau ( $n+1$ ) peut prendre différentes formes dérivées des règles de composition utilisées pour inclure les holons d'un niveau dans ceux d'un niveau supérieur.

Il est à noter que les holarchies ne sont pas des holons ni des systèmes physiques d'holons, mais ce sont des *arrangements conceptuels d'holons* qui représentent les *entités formelles de base* en vue d'une interprétation holonique des *structures* et *dynamiques* de la réalité.

La fonction d'holarchie à mettre en évidence réside dans les propriétés d'holons émergeant d'un niveau particulier de l'holarchie, et les formes d'auto-organisation qu'on ne pourra comprendre que par la connaissance des propriétés d'holons subordonnés formant sa structure, aussi bien que ceux des holons supérieurs dont ils sont des éléments structurels.



**Figure 3.4** – Modèle d'holarchie descendant (Source : [Mesarovic et al. (2000)])

Les concepts holon et holarchie sont devenus très répandus dans plusieurs disciplines particulièrement la psychologie cognitive [Young (1995)], les sciences physiques, l'ingénierie, la biologie [Ulieru and Geras (2002), Leitao (2009)], la gestion, le management, l'entreprenariat, le business [Malviya (2017)], l'industrie [Van Brussel et al. (1998)], et la robotique [Arai (1997), Sugi et al. (2003)], ainsi que la planification des transports [Bürckert et al. (2000)], et les systèmes de productions flexibles [Maturana et al. (1999), McFarlane and Bussmann (2000), Babiceanu and Chen (2006), Indriago et al. (2016)].

### 3.3. Les systèmes multi-agents (SMA) et l'approche holonique

L'étude de tout système ou environnement se base essentiellement sur les aspects structurel, dynamique et fonctionnel [Rumbaugh et al. (1991)], permettant d'une part de connaître l'évolution de sa structure, et d'autre part de définir les activités correspondantes. Soit à travers une vision macroscopique qui s'occupe des composants de grande taille d'un système et de leur fonctionnement global, ou bien par une vision microscopique qui s'intéresse aux composants élémentaires, l'intérêt revient à mieux comprendre le système, à cerner ses problèmes, et à proposer des solutions adéquates.



Dans ce contexte, plusieurs approches ont été proposées pour modéliser les systèmes du monde réel (biologiques, physiques, sociaux, économiques, industriels, etc.), et particulièrement les systèmes distribués à grande échelle qui manipulent des connaissances hétérogènes. Parmi d'autres l'approche multi-agents comme étant une technologie bien adaptée à l'étude de la dynamique des déplacements et d'actions des individus, des objets, ou des systèmes dans l'environnement.

### 3.3.1. Les SMA caractéristiques, avantages et limites

Les systèmes multi-agents (SMA) demeurent un domaine de recherche très actif, en connexion avec plusieurs domaines en particulier l'IA, le génie logiciel et les nouvelles technologies de l'information. Ils constituent une alternative pour la conception, la mise en œuvre ou la simulation et la compréhension de systèmes coopératifs, distribués et ouverts [Phan and Varenne (2010)]. Ce sont des organisations orientées-objectif d'agents intelligents, qui interagissent ensemble à la place des utilisateurs [Wooldridge (2009)]. Ils concrétisent en quelque sorte les entités ou les individus du monde réel [Duffey and Stratford (1989)], et facilitent le processus décisionnel qui nécessite de réagir rapidement même dans des situations complexes.

De tels systèmes s'intéressent aux comportements collectifs émanant des interactions entre des entités flexibles et autonomes, par la coopération et la concurrence afin d'assurer leurs coexistences. Ce qui caractérise principalement les SMA ce n'est pas seulement le partage ou la distribution des connaissances et ressources communes, mais aussi de faire coopérer un ensemble d'agents, de coordonner leurs actions, et de négocier entre eux [Weiss (1999)] afin de réaliser une activité collective cohérente.

D'après Weiss [Weiss (1999), Saddem and Hmida (2014)] les SMA possèdent aux plupart des caractéristiques suivantes :

- *Distribution* : la modularité du système affecte à chaque agent sa tâche appropriée ;
- *Autonomie* : dans les SMA, les agents conservent leur autonomie ;
- *Décentralisation* : concerne en particulier les décisions prises par les agents du système.
- *Echange de connaissances* : les agents communiquent entre eux en utilisant des langages spécifiques ;
- *Interaction* : suite aux motivations locales, les agents se comportent dans un environnement à grade égale sans subir des ordres supérieurs ;
- *Organisation* : elle se base sur les relations permettant l'interaction entre les agents.
- *Etat* : il est défini par l'évolution de l'ensemble des agents en interaction dans leur environnement ;
- *Ouverture* : l'échange des informations avec le monde extérieur, c'est-à-dire les agents peuvent opérer avec d'autres SMA ;
- *Emergence* : l'interaction des entités locales dotées chacune de ses propres spécifications mènent à des fonctions globales ;

- *Adaptation* : les agents dans un SMA adaptent instantanément leur comportement vis-à-vis l'environnement ;
- *Délégation* : afin de supporter la complexité dans la prise de décision, il est possible de déléguer le pouvoir à des agents pour assurer une partie du contrôle global de l'application ;
- *Personnalisation* : dans un SMA appartenant aux systèmes intégrés, les agents utilisateurs doivent s'adapter à ce système ;
- *Intelligibilité* : c'est de rendre les SMA plus accessibles par l'utilisateur extérieur.

Les SMA sont des systèmes modulaires, conçus principalement pour résoudre des problèmes concurrents dans des environnements distribués, qui offrent plusieurs avantages [Wooldridge and Jennings (1995)] :

- La facilité d'implémenter les problèmes dont les données et le contrôle sont distribués ;
- La vitesse, la fiabilité et la robustesse (le système continue à fonctionner cas d'échec d'un agent) ;
- La réutilisabilité et l'extensibilité du système en réintégrant les agents dans de nouveaux SMA et en ajoutant de nouveaux agents.

La plupart des recherches récentes focalisent beaucoup plus sur l'application des SMA que sur la solution théorique, allant des systèmes simples d'aide à la décision, jusqu'au systèmes ouverts, complexes et systèmes critiques pour les applications industrielles [Dorri et al. (2018)].

Par leur potentiel dans le domaine industriel, les SMA ont montré leur utilité dans différentes applications notamment le processus de control [Jennings (1995)], le système des diagnostics [Albert et al. (2003)], l'industrie manufacturière [Peng et al. (1998)], et la gestion des réseaux [Bieszczad et al. (1998)].

La nature distribuée et le volume important d'informations font d'Internet un domaine idéal pour les SMA dans la gestion des informations [Decker et al. (1997)], en mettant en œuvre des agents pour la recherche et le filtrage des informations [Klusch (2001)], Ce qui a permis d'introduire la technologie SMA dans d'autres domaines comme le commerce électronique, le processus d'affaires automatisé [Teo et al. (2012)], la gestion du trafic urbain, les logistiques de transport [Benaissa (2013)]. D'autres applications ont montré l'efficacité de cette technologie, par exemple les systèmes ubiquitaires [Salazar et al. (2015)], les réseaux sensoriels dense et larges pour le contrôle des paramètres comme pour les applications de l'environnement [Athanasiadis and Mitkas (2004)].

Les SMA possèdent tellement d'avantages et particulièrement dans le domaine des systèmes distribués qu'ils ont des défis à relever en termes de conception et d'implémentation [Sycara (1998)], notamment le problème de décomposition, la communication, la coordination globale, les solutions technologiques, et la prise de décision.

Dans la pratique, les SMA se montrent plus flexibles par rapport aux modèles macroscopiques, et particulièrement quand il s'agit de simuler les sociétés et les phénomènes spatiaux et évolutifs [Wooldridge (2009)]. C'est pour cela, qu'il est important de trouver un modèle d'organisation hiérarchique permettant de gérer des systèmes complexes à grande échelle, et sert à décomposer le système étudié, comme dans le cas des systèmes holoniques, de façon à ce que tous les composants soient traités en partie et réalisant un objectif commun.

### 3.3.2. L'agent holonique

En associant l'approche multi-agents et le concept d'holon dans une organisation hiérarchique, celle-ci constitue un système multi-agents holonique (SMAH), possédant une architecture « holarchique », dont les composants sont des holons jouant le rôle des agents, ayant des propriétés plus intéressantes, et étant à la fois une entité et une organisation [Rodriguez (2006)], tout en respectant les règles de Koestler relatives aux systèmes holoniques [Koestler (1969)] ainsi que celles définies par l'organisation qui doit s'adapter à l'environnement.

Ces agents holoniques autonomes et dépendants des autres agents holoniques s'ordonnent hiérarchiquement dans une organisation, offrant à chaque agent sa position à la quelle dépendent ses connaissances et ses actions, et le permettant de choisir librement sa stratégie dans son environnement.

Ce comportement s'appuie sur la communication par des messages entre les agents et leurs supérieurs ou bien entre les agents de la même couche. Ils sont de plus en plus cognitifs<sup>13</sup> vers le haut de l'holarchie et de plus en plus réactifs<sup>14</sup> vers le bas.

En fait, chaque agent holonique est reconnu par son identité, son état, ses connaissances, et son comportement dans son environnement et envers les autres holons [Adam et al. (1999)].

En tenant compte des notions d'autonomie et de coopération, l'organisation holonique doit s'adapter à son environnement, où les parties opèrent individuellement ou coopèrent pleinement.

En comparant l'agent holonique à l'agent intelligent, on distingue une différence explicite sur les points suivants [Calabrese et al. (2010)] :

- *L'information et le traitement physique* qui sont présents dans les holons tandis que les agents sont seulement considérés comme des entités de programmes ;
- *La récursivité* qui caractérise particulièrement les holons, mais ce n'est pas le cas pour les agents ;

---

<sup>13</sup> Un *agent cognitif* dispose généralement d'une base de connaissances nécessaires pour accomplir sa tâche, effectuer son interaction avec les autres agents, ainsi que les objectifs pour prendre ses décisions.

<sup>14</sup> Tandis qu'un *agent réactif* acquiert des perceptions pour agir et réagir rapidement aux changements éventuels de son environnement simplement par l'application des règles prédéfinies. L'agent peut avoir à la fois un aspect cognitif (délibératif) et un aspect réactif, on peut donc définir un *agent hybride*.

- *L'organisation* dont les holons s'organisent suivant des holarchies, qui sont généralement représentées comme étant des structures hiérarchiques dynamiques [Zhang and Norrie (1999a), Zhang and Norrie (1999b)], tandis que les architectures agent sont statiques et s'ordonnent d'une organisation horizontale vers une organisation verticale [Sycara (1998), Okamoto et al. (2008)].

En se référant à l'analyse faite par Mark et Pechoucek [Mark et al. (2002)], qui a été présenté par Giret et Botti [Giret and Botti (2004)], la comparaison des caractéristiques entre un agent intelligent et un agent holonique est résumée dans le tableau suivant (Tableau 3.1) :

<i>Caractéristiques</i>	<i>Agent</i>	<i>Holon</i>
Autonomie	Oui	Oui
Réactivité	Oui	Oui
Proactivité	Oui	Oui
Capacité sociale	Oui, l'interface utilisateur est implémentée par un ou plusieurs agents spécialisés	Oui, l'interface utilisateur est spécifique à chaque holon
Coopération	Oui, il peut être compétitif et coopératif	Oui, les holons ne rejettent pas exprès la coopération avec un autre holon
Organisation, ouverture	Oui, hiérarchies, organisations horizontales, hétérarchies, etc. Les holarchies sont implémentées en utilisant différentes approches d'architecture SMA pour des fédérations comme les facilitateurs, les courtiers (brokers), ou les médiateurs...	Oui, holarchies
Rationalité	Oui	Oui
Apprentissage	Oui	Oui
Bénévolat	Oui	Oui
Mobilité	Oui	Holons nécessitent rarement la mobilité pour l'exécution de ses tâches.
Récurtivité	Il n'y a pas d'architecture récursive, néanmoins d'autres techniques pourraient être utilisées pour définir des groupes simulant les différents niveaux récursifs	Oui
Traitement des informations et traitement physique	Il n'y a pas de séparation explicite	La séparation est explicite pourtant le traitement physique est optionnel
Attitudes mentales	Oui	Ils n'ont pas besoin de raisonner par leurs propres attitudes mentales ou par celles des autres unités de contrôle

**Tableau 3.1** – Holon vs Agent [Giret et Botti, 2004]

Sachant que l'autonomie et l'auto-organisation sont deux propriétés distinctes des agents intelligents et des SMA respectivement, l'agent holonique est principalement fondé comme un objet modulaire permettant de renforcer efficacement la modélisation et le développement, disposant des caractéristiques importantes issues du génie logiciel.

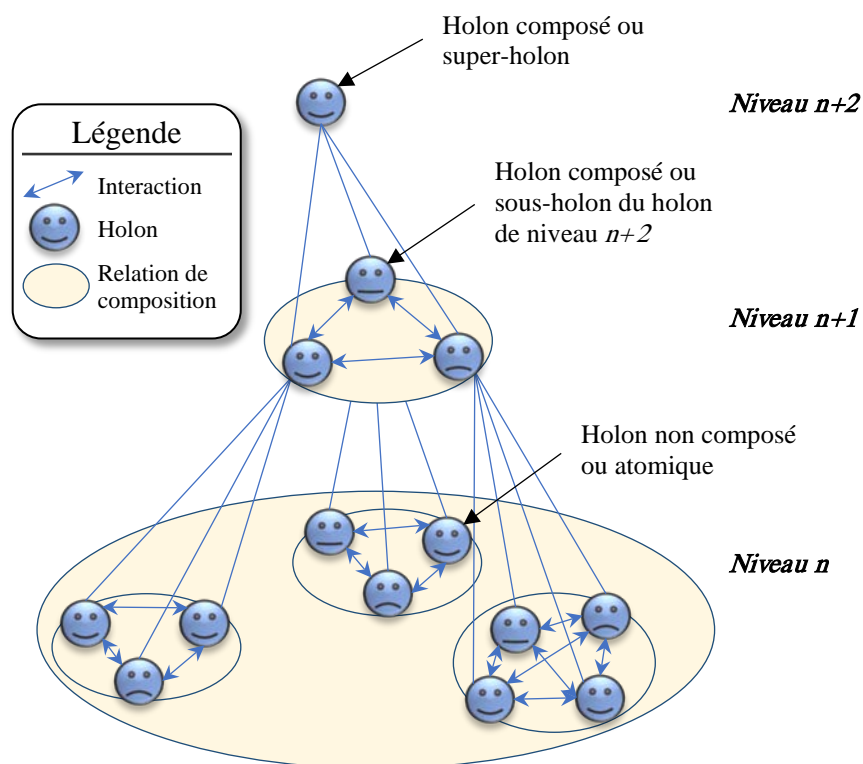
### **3.4. Les systèmes multi-agents holoniques (SMAH) et leurs applications :**

Généralement, les agents dans les SMA sont des entités atomiques, il s'agit donc de trouver un modèle capable de représenter un groupe, une organisation ou une société d'agents, pouvant se joindre à la structure d'un super agent et d'agir comme étant une seule entité.

Dans ce sens, plusieurs travaux de recherche ont employé différents noms spécifiques pour designer l'idée de composition d'un agent de plusieurs agents [Bendakir (2016)], tels que : méta-agents [Holland (1995)], agents intermédiaires [Courdier et al. (1998)], agents collectifs et individuels [Ferber and Weiss (1999)], les SMA holoniques [Gerber et al. (1999a)] et les agents récursifs ou intermédiaires [Fernandes and Christine (2001)].

Etant adopté par la *communauté de l'intelligence artificielle distribuée* [Gerber et al. (1999a)], les systèmes multi-agents holoniques (SMAH) font partie des systèmes multi-agents adaptatifs, dont le comportement des agents et/ou la structure de l'organisation change afin de s'adapter et d'être compatible à l'environnement en vue de réaliser une fonction. Ces systèmes possèdent des propriétés émergentes et des capacités d'apprentissage permettant d'adopter un comportement approprié. Pourtant chaque agent autonome réalise une fonction partielle, le résultat du système dépend fortement de son organisation pour accomplir une fonction globale.

Dans la théorie des SMAH, la composition d'agents holoniques (Figure 3.5) – les agents sont composés d'agents et s'assemblent pour former d'autres agents – est issue d'une inspiration abstraite pour les systèmes hiérarchiques ayant un comportement intelligent ; en plus ils permettent une modélisation des phénomènes complexes.



**Figure 3.5** – Composition des agents holoniques dans un SMAH  
(Source : [Gaud (2007)]).

Depuis son apparition, l'approche holonique n'a cessé d'évoluer, et conjointement aux SMA a donné naissance aux dits SMA holoniques comme un nouveau paradigme de

modélisation. En outre les agents holoniques sont avérés bénéfiques dans divers domaines, on peut citer l'industrie manufacturière, les systèmes d'information médicaux, système de gestion à distance des entreprises de transport [Gaud (2007)], la logistique, la fabrication flexible, la production, et l'enseignement à distance [Gerber et al. (1999b), Bürckert et al. (2000), Maturana et al. (1999), Van Brussel et al. (1998), Ulieru and Geras (2002)].

### **3.5. Application de l'approche holonique à la planification des projets urbains**

La complexité du système urbain constitue une réalité qui évolue constamment, ce qui nécessite l'adoption de nouvelles méthodes, démarches et outils de planification et d'aide à la décision, permettant aux pouvoirs publics et privés de prévoir, d'orienter, d'organiser et de maîtriser le développement du territoire urbain, tout en impliquant l'ensemble des acteurs concernés par le processus de la planification urbaine.

L'analyse du territoire urbain et le diagnostic de l'état présent facilitent l'élaboration des meilleurs plans urbains, portant les actions à entreprendre afin d'assurer une bonne répartition des composantes urbaines. En effet, le plan urbain projette à l'échelle temporelle les opérations urbaines à concrétiser sur le territoire.

De fait, il est primordial d'utiliser une approche adéquate permettant d'observer les différents niveaux hiérarchiques du système urbain à partir du niveau local jusqu'au niveau global.

Ce chapitre présente la démarche proposée pour la mise en œuvre d'un système intelligent d'aide à la décision pour la planification des projets urbains en employant l'approche holonique comme étant un paradigme de modélisation offrant une décomposition cohérente et flexible des systèmes complexes et à grande échelle. Notre contribution couvre également la conception des agents holoniques constituant notre système multi-agents holonique, ainsi que l'implémentation d'un algorithme méta-heuristique inspiré de l'algorithme de fourmis multi-objectifs dont le rôle est d'optimiser le processus de la planification des projets urbains.

### **3.6. Aperçu sur le système urbain**

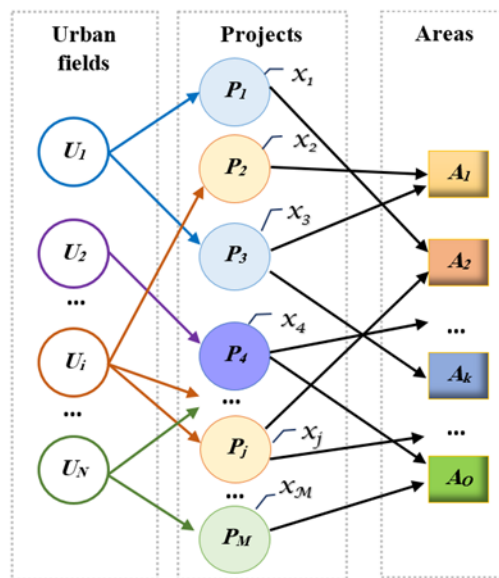
Le système urbain est un organisme entier, défini par ses propres caractéristiques dont les composantes sont réparties sur des zones urbaines (Figure 3.6), issues d'un partitionnement selon des facteurs géographiques, socio-économiques et administratifs.

Ce principe de zoning simplifie au décideur d'entreprendre les actions nécessaires pour satisfaire les besoins du citoyen, cependant la gestion des zones séparément ne permet pas d'atteindre les objectifs communs du territoire.



fonctions essentielles sont l'acquisition des données, l'analyse des données, le traitement et la visualisation des données. Il s'agit donc de déterminer les besoins du territoire urbain, ce qui aide les décideurs à choisir les projets urbains dans différents secteurs d'activité afin de les affecter aux zones appropriées (Figure 3.8).

Les urbanistes, les architectes et les décideurs travaillent ensemble dans le but de trouver les meilleures solutions pour l'intérêt public. Dans ce contexte, la planification du projet urbain consiste à sélectionner un ensemble de projets de développement pour promouvoir une zone urbaine selon les besoins exprimés. De ce fait, les projets urbains sont perçus comme une agrégation d'unités élémentaires, par exemple dans l'éducation l'unité élémentaire est la classe et dans le secteur de la santé, un lit d'hospitalisation correspond à une unité élémentaire. Par conséquent, il est nécessaire de mettre à la disposition autant d'unités de projets qu'une zone urbaine en a besoin, mais ce n'est pas possible particulièrement en cas de manque ressources.



**Figure 3.8** – Problème d'affectation de projets urbains par secteur d'activité aux zones urbaines (Source : [Khelifa et al. (2018)])

L'organisation en multicouches se montre efficace pour manipuler des systèmes urbains simples dont les zones et les secteurs d'activités sont limités, en revanche il devient difficile de supporter tous les détails des composantes urbaines pour des systèmes complexes et à grande échelle. De fait, il est primordial de recourir à une approche appropriée permettant une décomposition cohérente et flexible du système urbain, pour mieux observer les différents niveaux hiérarchiques à partir du niveau local jusqu'au niveau global, et de garantir l'intégrité du système.

### 3.7. Vers une organisation holonique de la planification des projets urbains

Le système urbain réunit des infrastructures réparties sur les zones urbaines selon différents secteurs d'activité. Les autorités locales œuvrent de manière autonome pour



promouvoir les projets urbains et collaborent ensemble pour réaliser des objectifs communs.

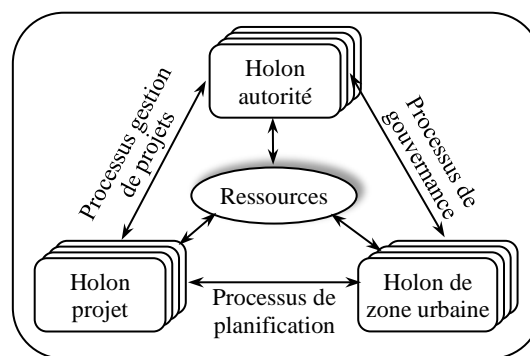
Il s'agit de regrouper les zones adjacentes à partir d'un niveau bas et les relier avec d'autres zones du même niveau pour joindre un niveau supérieur et ainsi de suite. L'aperçu de ce système tend vers une architecture holonique. Dont les holons similaires sont interconnectés à travers l'inclusion dans différents niveaux formant une holarchie.

Partant de la vision holonique des systèmes multi-agents, les holons sont organisés d'une manière récursive, qui signifie que chaque holon peut contenir des niveaux inférieurs d'holons, qui sont à leur tour contenus dans un autre niveau supérieur d'holons.

Par la projection de l'architecture holonique sur le système urbain, on distingue les holons suivants :

- Les holons du territoire urbain qui peuvent être divisés en zones urbaines, divisées elles-mêmes en sous-zones et ainsi de suite.
- Les holons d'autorité identifient la relation de pouvoir qu'une autorité d'un niveau supérieur comprend les autorités subordonnées, qui à leur tour contiennent des autorités subordonnées, etc.
- La troisième catégorie indique les holons de projet urbain qui peuvent être divisés en sous-projets, puis en sous-sous-projets jusqu'à atteindre les unités élémentaires de projet urbain.

Dans le domaine de la planification des projets urbains, ces holons partagent des ressources communes en formant un réseau holonique structuré comme un système multi-agents holonique (SMAH) cf. voir (Figure 3.9).



**Figure 3.9** – Vue globale du réseau SMAH de la planification des projets urbains  
(Source : [Khelifa and Laouar (2020)])

### 3.8. Modélisation du SMAH de la planification des projets urbains

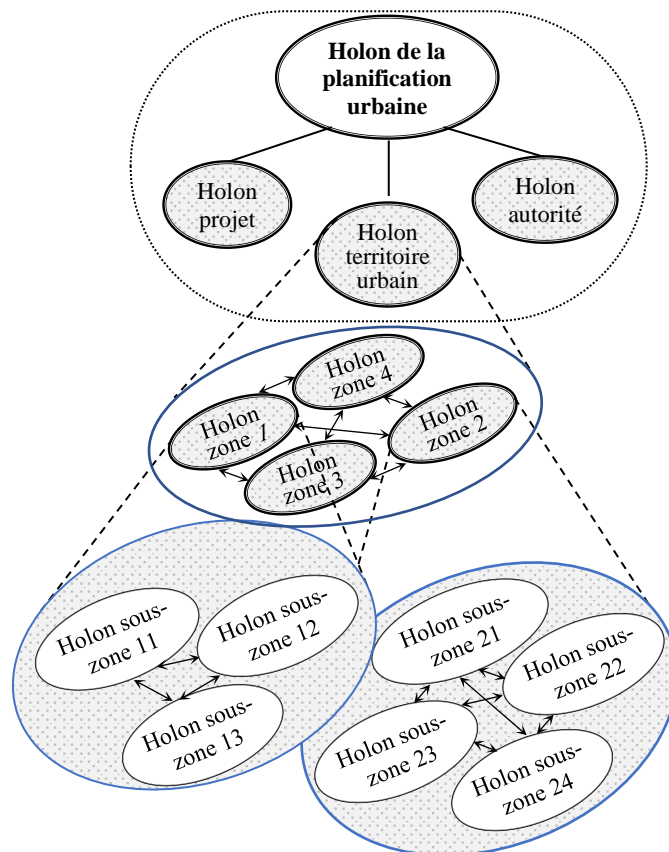
En tenant compte de la répartition administrative et du territoire urbain, l'organisation holonique sert à modéliser un réseau composé d'un ensemble de zones urbaines. Sous le contrôle des agents, chaque zone urbaine est composée de plusieurs sous-zones adjacentes. Les projets appartenant aux zones urbaines ainsi que leurs autorités sont groupés pour former un holon supérieur appelé l'holon de la planification urbaine, et

contrôlé par un agent supérieur de prise de décision. Il en résulte une holarchie à trois niveaux permettant de modéliser ce réseau comme illustré dans la figure (Figure 3.10).

Au niveau plus bas, dans chaque sous-zone un ensemble d'agents de planification interagissent comme étant des fourmis en appliquant l'algorithme d'optimisation par colonies de fourmis pour satisfaire des objectifs locaux par l'élaboration des plans de projets urbains optimaux formés d'unités provenant différents types de projets.

Pour le même intérêt, plusieurs sous-holons joignent en une structure cohérente d'un holon supérieur, qui à son tour est groupé avec d'autres holons du même niveau émergeant ensemble en une seule top entité appelée holon de planification urbaine, dont la tâche principale est de tracer la stratégie globale à appliquer par les sous-holons.

La structure récursive permet simultanément aux holons de contenir d'autres holons de niveau inférieur et d'être contenus dans un niveau supérieur d'holons.



**Figure 3.10** – L'architecture holonique proposée pour le système de planification des projets urbains (Source : [Khelifa and Laouar (2020)])

Formellement, le système holonique proposé pour la planification des projets urbains peut être défini par le quadruplet  $(H, \Omega, \Psi, \mathcal{F})$ :

- $H = \{h_1, h_2, \dots, h_z\}$ : est un ensemble fini d'holons, où chaque holon  $h_i$  possède une tête et un corps (sous-holon), dans lequel les  $M_i$  agents (*fourmis*) en interaction  $\in A = \{a_1, a_2, \dots, a_{M_i}\}$  collaborent pour établir les plans des projets urbains.

Notons que la fonction  $head(h_i)$  retourne la tête et la fonction  $subholon(h_i)$  retourne le corps du holon  $h_i$ . Pour les holons du niveau plus haut  $head(h_i) = h_i$ .

Les holons sont organisés sous une structure multi-niveau structure  $l = (l_0, l_1, \dots, l_{max})$ . Conventionnellement le premier niveau de planification ( $l_0 = 0$ ) comprend les holons de base des agents atomiques, et  $l_{max}$  définit l'holon top du haut niveau des agents de la prise de décision, quant au holons intermédiaires leur niveau est donné par fonction récursive :  $level(h) = level(subholon(h)) + 1$ .

Les holons des niveaux bas, milieu, et top sont déterminés respectivement par leurs vecteurs de variables de décision  $X, Y, Z$ . En respectant l'holarchie, les variables du niveau de base  $X$  sont groupés en variables  $Y$  du niveau intermédiaire, qui sont consolidés au niveau top formant les variables  $Z$  (Figure 3.11). Plus le flux des solutions s'approche du niveau top, plus la dimension des vecteurs des variables décisionnelles diminue, donc il devient facile de vérifier les objectifs globaux et d'accélérer le processus d'optimisation.

Pour chaque holon de base  $h_i \in H \mid level(h_i) = 0$ , on définit le vecteur  $X_{h_i}$  de  $m_i$  variables de décision discrètes  $X_{h_i} = (x_1, \dots, x_{m_i})$ , tel que chaque variable  $x_j$  représente le nombre des unités sélectionnées d'un projet urbain  $p_j \in P, P = \{p_1, \dots, p_{m_i}\}$ , qui peut avoir une valeur appartenant au domaine  $d_j = (u_j^1, \dots, u_j^{|d_j|}), j = 1 \dots m_i$ .

Les composants fondamentaux formant tout projet  $p_j$  sont des unités similaires, où chaque unité élémentaire est définie par un ensemble de caractéristiques pondérées  $\mathbb{C}_j = \{c_j^1, \dots, c_j^{|\mathbb{C}_j|}\}$  utiles pour évaluer les préférences  $v$  et les coûts  $c$  du projet. En combinant toutes les valeurs possibles de chaque variable décisionnelle  $x_j$  d'un holon  $h_i$ , le résultat est la séquence de solutions candidates  $S_{h_i} = \langle S_{h_i}^1, S_{h_i}^2, \dots \rangle$  des plans de projets urbains au niveau local.

- $\Omega$  est l'ensemble des contraintes sur les variables décisionnelles du système holonique, selon lequel chaque holon  $h_i$  est soumis à un sous ensemble de contraintes  $\Omega_{h_i} = \left\{ g_{h_i}^1, \dots, g_{h_i}^{|\Omega_{h_i}|} : g_{h_i}^k \leq 0, k = 1..|\Omega_{h_i}| \right\}$ .

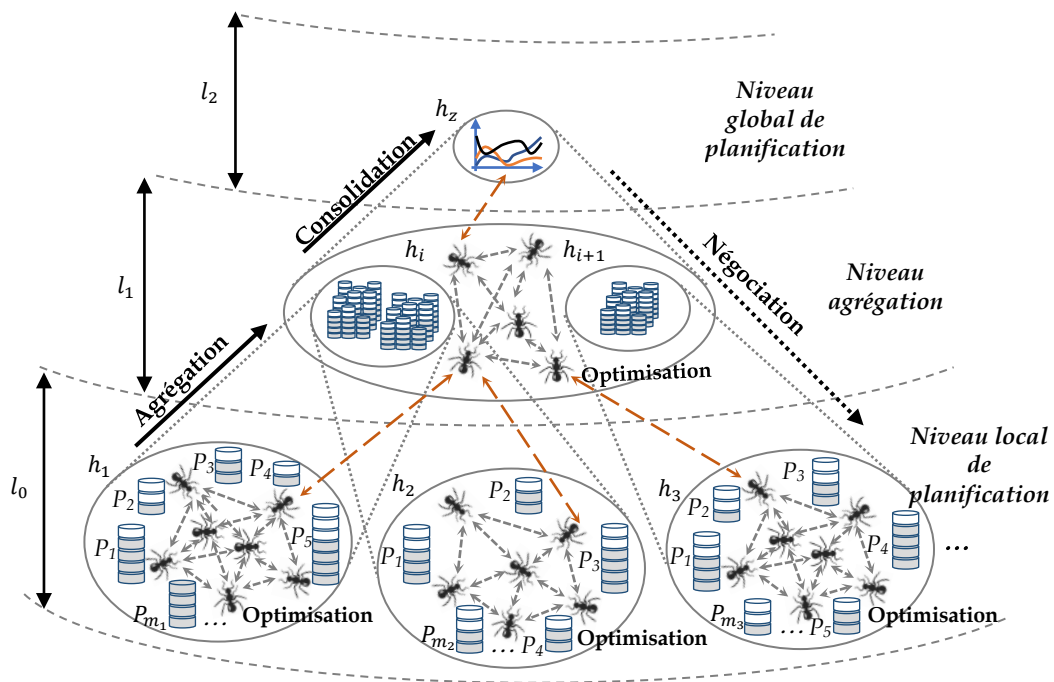
Afin d'éviter les mauvaises décisions, il est fortement recommandé de bien formuler les contraintes en question, et de les pousser vers les niveaux bas du système holonique, dans le but d'ignorer les solutions infructueuses et accélérer ainsi la résolution du problème d'optimisation.

- $\Psi = (S_1, \dots, S_n)$  est l'espace de toutes les solutions faisables formées à partir d'un ensemble fini de  $m$  variables décisionnelles discrètes  $x_j$  provenant de tous les holons de base,  $j = 1..argmax(m_i)$  via l'ensemble des contraintes  $\Omega$ . Ces solutions basiques constituent les plans de projets urbains au niveau local pour construire les plans globaux urbains résultants.

Quand le processus d'optimisation s'arrête sans retourner aucune solution satisfaisante, les décideurs du haut niveau entament un processus de négociation bilatérale avec ceux du niveau inférieur en vue d'un consensus à travers la réallocation des ressources limités et/ou même revoir les besoins en matière de projets urbains.

- $\mathcal{F}$  vecteur de  $r \geq 2$  fonctions objectif,  $f: \Psi \rightarrow \mathbb{R}, \mathcal{F}(S) = (f_1(S), \dots, f_r(S))$ ; sans perte de généralité, chaque fonction  $f$  doit être optimisé selon son objectif.

L'optimisation multi-objective vise à satisfaire en quelque sorte toutes les fonctions objectifs sous l'ensemble de contraintes  $\Omega$ , on doit trouver une solution optimale globale  $S^*$  parmi les solutions faisables, qui par convention minimise la fonction objectif global  $\mathcal{F}$ , tel que  $\mathcal{F}(S^*) \leq \mathcal{F}(S) \forall S \in \Psi$ , dont l'ensemble est noté par  $\Psi^* \subseteq \Psi$ . Dans ce système holonique, l'optimalité globale s'appuie sur les plans optimaux du niveau local développés à travers les différents niveaux comme étant des solutions produites par les agents. Les agents de chaque holon communiquent ensemble par une stratégie de coopération, mais les holons du même niveau ne peuvent être interconnectés qu'à travers les holons du niveau supérieur, pour que chacun d'eux reconnaisse la stratégie des autres.



**Figure 3.11** – Système holonique de la planification des projets urbains  
(Source : [Khelifa and Laouar (2020)])

Pour réaliser un maximum de profit, chaque holon de base applique sa propre stratégie afin de choisir de manière compétitive des unités appartenant aux différents projets. C'est de faire le mieux pour lui-même et pour les autres. Théoriquement, ceci rejoint le principe d'équilibre de Nash [Nash (1951), Canning (1992), Bach and Perea (2020)] c'est-à-dire un holon ne pourrait pas choisir de stratégies autres que celles qui donnent les meilleurs

plans urbains globaux, et aucun holon de base ne pourra maximiser son propre gain en changeant sa stratégie, pendant que les autres holons maintiennent leurs stratégies inchangées.

Pour produire les plans des projets urbains, nous avons employé l'approche ascendante, dont les agents de planification déployés pour les différentes sous-zones urbaines utilisent un algorithme d'optimisation par colonies de fourmis pour optimiser les plans locaux, puis les communiquer aux niveaux supérieurs pour d'éventuels analyse et contrôle, et finalement au niveau de l'holon de la planification urbaine, l'algorithme opère pour achever les plans globaux.

Puisque les SMAH s'organisent dans une structure récursive, l'algorithme méta-heuristique d'optimisation multi-objectifs par colonies de fourmis (*holonic multi-objective ant colony optimization* H-MACO) à implémenter devrait supporter la récursivité pour assurer la tâche de la planification urbaine décentralisée.

L'algorithme H-MACO adapté suit un processus ascendant en appelant récursivement à chaque holon de sous-zone l'algorithme appelé BKPACS, inspiré de l'algorithme basique d'optimisation par colonies de fourmis (*ant colony system* ACS) et modifié pour résoudre le problème de la planification des projets urbains modélisé en tant qu'un problème du sac-à-dos borné (*bounded knapsack problem* BKP).

### 3.9. La planification des projets urbains en tant que BKP

Un plan urbain saisit rationnellement une sélection d'unités de différents types de projets en vue de réaliser le maximum de profit sans dépasser les limites de ressources. Ceci tend d'être représenté comme un problème du sac-à-dos borné. Donc le problème de la planification des projets urbains est défini comme suit :

Un type d'élément est un projet  $p_i$  parmi  $n$  projets ; le poids est le coût de l'unité  $c_i$  ; le profit est la valeur de préférence  $v_i$  qui indique le degré d'importance du projet  $p_i$  ; et la capacité du sac-à-dos est définie par le budget alloué  $B$ , limitant le total des coûts des unités sélectionnées dans un plan.

L'essentiel, c'est de saisir à partir de chaque projet  $p_i$  un nombre d'unités  $x_i$  dont la borne supérieure est  $u_i$ , pour que la somme des coûts ne dépasse pas le budget alloué  $B$ , et le total des préférences soit maximisé.

Le problème du sac-à-dos de la planification des projets urbains est alors formulé par :

$$\left\{ \begin{array}{l} \max \sum_{i=1}^n v_i x_i \\ \text{s. c.} \sum_{i=1}^n c_i x_i \leq B \\ 0 \leq x_i \leq u_i, x_i \text{ entier}, \forall i \in \{1, \dots, n\} \end{array} \right. \quad (1)$$

Le problème du sac-à-dos borné est connu par ses coefficients entiers positifs : la valeur de préférence  $v_i$ , le coût d'unité  $c_i$ , la borne  $u_i$  appliquée sur le nombre d'unités

élémentaires, et la borne supérieure  $B$  qui limite le total des coûts des unités des projets sélectionnés.

Le problème du sac-à-dos borné est considéré comme une généralisation du *non polynomial-difficile* (NP-difficile) problème du sac-à-dos 0-1 (0-1 KP). Généralement, le BKP est transformé en 0-1 KP, qui pourrait être résolu en un temps pseudo-polynomial par plusieurs méthodes comme la programmation dynamique ou la méthode séparation-évaluation progressive [Silvano and Paolo (1990), Andonov et al. (2000)]. Ces algorithmes s'appuient sur des coefficients entiers pour définir les dimensions des tables et les paramètres des fonctions ; cependant dans la pratique comme pour le cas de la planification des projets urbains, et sans transformer les valeurs des coefficients réels en valeurs entières, ces algorithmes ne sont pas applicables.

Le BKP dédié à la planification des projets urbains semble être un peu différent, dans la mesure où les éléments ne sont pas homogènes, et possèdent des valeurs de préférences incertaines ainsi que des coûts écartés. D'autant plus lors de la recherche d'une solution optimale, ces algorithmes dérivent des solutions réalistes. Les éléments sont triés dans un ordre décroissant des valeurs d'efficacité  $e_i = v_i/c_i$ , ainsi la solution optimale portera le maximum d'unités de la liste des éléments ordonnés, tout en commençant par le premier élément et allant directement jusqu'à ce que le budget soit épuisé ; malheureusement, les éléments restants en fin de liste ne sont pas pris en compte même s'ils sont plus sollicités que les autres.

Malgré l'efficacité de ces algorithmes pour trouver des solutions optimales en un temps réduit, les plans générés restent faibles en matière d'équilibre et de diversité. En fait, les bornes supérieures d'unités de projets représentent des valeurs rationnelles à prendre en compte, elles sont en quelque sorte des objectifs implicites. Par exemple, pour des mesures de préparation à une catastrophe ou un imprévu, les abris devraient être fournis même s'ils ne seront probablement pas utilisés.

La qualité d'une solution dépend fortement de la diversité des éléments retenus, il est donc préférable de saisir le maximum des types d'éléments. Pour chaque projet  $p_i$  l'hypothèse du BKP  $u_i \cdot c_i \leq B$ , peut être remplacée simplement par  $c_i \leq B$ , ce qui signifie chaque projet candidat  $p_i$  pourrait fournir au moins une unité de projet unique avec un coût  $c_i \leq B$  indépendamment de la borne  $u_i$ .

Comme il n'y a pas plus que  $\lfloor B/c_i \rfloor$  d'unités de projet  $p_i$  qui peuvent être retenues dans les limites du budget  $B$ , et pour éviter d'explorer des valeurs inutiles, la borne  $u_i$  pourrait être ajustée à une nouvelle valeur entière  $u'_i = \min(u_i, \lfloor B/c_i \rfloor)$ . Cependant, la borne initiale  $u_i$  reste toujours utile ainsi que le coût  $c_i$  et la valeur de préférence  $v_i$  pour évaluer l'efficacité des projets dans la fonction objectif.

Admettons que les éléments de projets à ordonner dans un ordre décroissant de leurs valeurs d'efficacité absolues  $e_i = u_i \cdot v_i / c_i$  telles que  $e_1 \geq e_2 \geq \dots \geq e_n$ . L'hypothèse  $\sum_{i=1}^n c_i \cdot u'_i > B$  exclut le cas trivial du problème lorsque le budget suffit à toutes les unités de projets nécessaires.

Les experts urbains recommandent les valeurs des préférences pour les projets urbains, qui sont éventuellement mises à jour dans le catalogue des caractéristiques de projets urbains<sup>15</sup> en fonction des spécificités de chaque zone urbaine. Ces préférences sont parfois incertaines, et leurs valeurs sont exprimées sur des échelles différentes en les comparant aux coûts, donc elles ne parviennent pas à caractériser la fonction objectif à optimiser sans y apporter les modifications appropriées. Pour rejoindre la même plage des coûts, il est nécessaire d'utiliser une méthode de normalisation comme la méthode *MinMax*, en mettant en correspondance respectivement les nouvelles valeurs minimales et maximales des préférences avec les valeurs minimales et maximales des coûts.

Au lieu de maximiser simplement le total des préférences des unités sélectionnées, le BKP de la planification des projets urbains cherche à maximiser la somme de leurs valeurs d'efficacité relatives qui sont évaluées sur la base des préférences normalisées. La valeur d'efficacité relative résultante est calculée par l'équation (2) :

$$e_i = \frac{\text{MinMax\_Normalized}(v_i)}{c_i} \quad (2)$$

Depuis plus d'un siècle, de nombreux travaux ont tenté de briser la difficulté du problème du sac-à-dos et de trouver la solution optimale dans le moindre temps, mais le KP ne peut pas être résolu exactement en un temps polynomial. Comme dans plusieurs problèmes d'optimisation combinatoire, les méthodes de résolution approchées y compris les approches heuristiques et méta-heuristiques qui donnent des solutions pseudo-optimales dans un temps raisonnable, sont donc les plus sollicitées [Tzeng and Huang (2013)].

Comme mentionné précédemment, la planification des projets urbains peut se ramener à un problème du sac-à-dos borné, qui est considéré comme un problème de décision NP-complet dont l'optimisation est également NP-difficile. Généralement, résoudre un problème de décision en un temps polynomial facilite l'optimisation du problème, en itérant l'algorithme de décision. D'autre part, s'il existe un algorithme polynomial pour trouver des solutions optimales, le problème pourrait être transformé en un problème de décision, et par la comparaison des solutions obtenues, on peut résoudre ainsi le problème de décision [Ausiello et al. (1995)].

En plus de sa complexité, le processus de la planification urbaine couvre de nombreux domaines repartis sur un étendu géographique important. Par conséquent, il est

---

<sup>15</sup> Le catalogue de projets urbains contient toutes les informations relatives aux projets, telles que le nombre d'unités, les préférences et les coûts concernant chaque zone urbaine.

nécessaire de faire appel à des approches et méthodes efficaces pour modéliser ce système et le décomposer en simples entités urbaines dans une organisation hiérarchique.

Grâce à ses concepts et ses règles, l'approche holonique se montre la plus appropriée. En effet, les entités urbaines indépendantes similaires interagissent ensemble sous forme d'holons. En remontant des niveaux bas aux niveaux supérieurs de la hiérarchie, le système urbain est modélisé comme un système multi-agents holonique (HMAS), c'est pour cela que nous avons développé un nouvel algorithme méta-heuristique d'optimisation par colonies de fourmis à déployer pour assurer la tâche des agents dont l'objectif est d'établir les plans urbains.

### **3.10. L'algorithme d'optimisation multi-objectifs par colonies de fourmis du système holonique**

Introduit par *Marco Dorigo* au début des années 90 pour résoudre le problème du voyageur de commerce (*Traveling Salesman Problem TSP*), l'algorithme méta-heuristique d'optimisation par colonies de fourmis (OCF) [[Dorigo et al. \(1996\)](#), [Dorigo and Gambardella \(1997\)](#)] inspiré du comportement des fourmis à la recherche de la nourriture, essayant de choisir le plus court chemin depuis la fourmilière jusqu'à la source de nourriture. Chaque fourmi dépose une substance chimique appelée « phéromone » pour marquer son itinéraire. Au long du chemin la phéromone pourrait s'évaporer, et par conséquent le chemin conservant une forte concentration de la phéromone est considéré optimal, ce qui attire les autres fourmis à le prendre.

L'algorithme utilise des fourmis artificielles pour imiter le processus de forage. Ces agents coopèrent et communiquent ensemble par la phéromone et partagent l'information heuristique. C'est en appliquant une stratégie stochastique, itérativement chaque fourmi tente de construire et mémoriser une solution complète reliant les nœuds depuis le point de départ à la destination.

Etant donné que le domaine urbain est un champ vaste, qui s'étend pour couvrir les différentes composantes urbaines, la planification globale des projets urbains devient très difficile et mène souvent à un problème d'explosion combinatoire. Donc, pour manipuler toutes ces composantes, il est recommandé d'employer l'architecture holonique qui divise le système urbain en holons.

A travers les niveaux mentionnés ci-dessus, commençant par le niveau inférieur de chaque sous-zone, les agents collaborent pour trouver les plans optimaux, qui seront consolidés au niveau supérieur pour une éventuelle évaluation. Le processus se répète pour obtenir de meilleures solutions faisables, qui seront à leur tour communiquées au holon de territoire urbain du plus haut niveau. Une fois le plan résultant satisfait l'objectif global, le processus d'optimisation se termine avec succès.

Dans chaque holon de base, les agents de planification utilisent l'algorithme BKPACS modifiant l'algorithme méta-heuristique *Ant Colony System (ACS)* pour résoudre le



problème du sac-à-dos borné adapté à la planification des projets urbains. Les sous-sections suivantes décrivent les éléments essentiels de cet l'algorithme.

### 3.10.1. L'information phéromonique

Dans ce contexte, la structure de l'information phéromonique (Figure 3.12) est différente de telle sorte qu'à chaque projet urbain  $p_i$  est associé un vecteur de phéromone  $\tau_i$  dont la dimension est la borne supérieure  $u_i$ , où chaque cellule  $\tau_{i,j}$  représente la désirabilité apprise de prendre un nombre d'unités  $x_i = j$  du projet courant  $p_i$ .

Durant le processus de construction de la solution, les fourmis modifient la quantité de la phéromone déposée sur les nœuds. En tenant compte de l'évaporation, la quantité restante est calculée par la formule suivante :

$$\tau_{i,j} \leftarrow (1 - \rho) \cdot \tau_{i,j} + \sum_{k=1}^M \Delta\tau_{i,j}^k \quad (3)$$

$\rho$  est le taux d'évaporation,  $M$  représente le nombre de fourmis, et  $\Delta\tau_{i,j}^k$  est la quantité de la phéromone sur le nœud qui correspond au nombre d'unités  $j$  du projet  $p_i$  sélectionnés par la fourmi  $k$ .

La quantité de la phéromone  $\Delta\tau_{i,j}^k$  est calculée par l'équation suivante :

$$\Delta\tau_{i,j}^k = \begin{cases} Q/L_k, & \text{Si } j \text{ unités sont sélectionnés du} \\ & \text{projet } p_i \text{ dans la solution } S_k \\ 0, & \text{Cas contraire} \end{cases} \quad (4)$$

Où, le constant positif  $Q$  est un paramètre pour normaliser les valeurs de la quantité de la phéromone,  $L_k$  est la longueur de la solution  $S_k$  construite par la fourmi  $k$ . Cette longueur représente combien est éloignée la solution par rapport à la limite, sa valeur est donné e par l'équation (3) :

$$L_k = \sum_{i=1}^n \frac{u_i - x_i}{u_i} \quad (5)$$

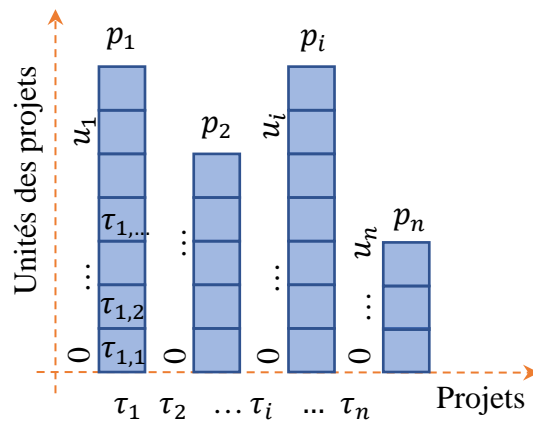


Figure 3.12 – La structure de la phéromone pour la planification des projets urbains (Source : [Khelifa and Laouar (2018)])

Dans le cas de projets avec un nombre important d'unités, il est préférable de regrouper ces unités en clusters similaires pour éviter le problème d'explosion de dimension. Par exemple les appartements peuvent être regroupés en blocs.

### 3.10.2. L'information heuristique

Pour ce problème du sac-à-dos borné, maximiser le profit, l'information  $\eta_{i,j}$  heuristique représente à quel point, il est avantageux de sélectionner un nombre  $j \leq u_i$  d'unités élémentaires du projet courant  $p_i$ . La valeur heuristique  $\eta_{i,j}$  est évaluée en utilisant la valeur d'efficacité relative  $e_{i,j} = j \cdot v_i/c_i$ , mentionné précédemment (section 3.9). Cette valeur heuristique est calculée par l'équation :

$$\eta_{i,j} = \begin{cases} j \cdot v_i/c_i, & \text{if } j \leq \min(u_i, [U_i]) \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

$[U_i]$  indique le plus grand nombre entier non supérieur à  $U_i$ . Il représente le nombre maximum d'unités du projet  $p_i$  pouvant être sélectionnées en utilisant le reste du budget  $B$ . La valeur de  $U_i$  est calculée par l'équation (7) :

$$U_i = \frac{B - \sum_{l < i} x_l \cdot c_l}{c_i} \quad (7)$$

### 3.10.3. La sélection probabiliste d'unités de projet

Au lieu de trier de façon décroissante les éléments par le rapport de préférence sur le coût ( $v_i/c_i$ ) comme dans la méthode gourmande (*greedy method*) proposée par Dantzig [Dantzig (1957)], les projets sont triés par valeur d'efficacité  $e_i = u_i \cdot v_i/c_i$  telle que  $e_1 \geq e_2 \geq \dots \geq e_m$ . Cela permet à l'agent de planification de construire progressivement des solutions, c'est-à-dire que chaque agent de planification se déplace le long des projets triés (Figure 3.12.), et sélectionne avec probabilité  $P_{i,j}^k$  un nombre d'unités  $j \leq u_i$  de chaque projet  $p_i$ , ce qui consomme par conséquent la valeur du budget  $B$ . La probabilité  $P_{i,j}^k$  est définie par l'équation suivante (8):

$$P_{i,j}^k = \begin{cases} \frac{\tau_{i,j}^\alpha \cdot \eta_{i,j}^\beta}{\sum_{l \in \mathcal{N}_i} \tau_{i,l}^\alpha \cdot \eta_{i,l}^\beta}, & \forall j \in \mathcal{N}_i \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

$\mathcal{N}_i$  est le nombre d'unités autorisé du projet  $p_i$ , que l'agent de planification peut sélectionner en utilisant la valeur restante du budget.

$\alpha$  et  $\beta$  sont des paramètres pour ajuster respectivement, la quantité de la phéromone et la valeur heuristique dans la probabilité de sélection d'unités de projets.

### 3.10.4. L'algorithme BKPACS

Un tel algorithme construit des solutions pseudo-optimales dans un temps de calcul réduit et supporte une fonction d'optimisation multi-objectifs qui maximise la somme les préférences des projets urbains, minimise les coûts, et converge vers les bornes (indiquant le nombre d'unités demandées de chaque projet), tout en assurant une diversité de projets sélectionnés, ce qui augmente la qualité des plans urbains. Les étapes de l'algorithme BKPACS sont les suivantes :

**Etape 1 :** chargement des informations des projets : attributs (préférences et coûts) et bornes (unités nécessaires). Normaliser les valeurs des attributs à l'aide de la méthode *MinMax-Normalisation*, puis trier ces projets dans un ordre croissant de leur valeur d'efficacité.

**Etape 2 :** initialiser tous les paramètres, y compris le nombre d'agents de planification (fourmis)  $M$ , le facteur de phéromone  $\alpha$ , le facteur heuristique  $\beta$ , le taux d'évaporation  $\rho$ , la constante de phéromone  $Q$ , le nombre d'itérations minimal  $MinIter$ , le nombre d'itération maximal  $MaxIter$ , l'information phéromonique  $\tau$ , l'ensemble des solutions  $S$  initialisé à  $\emptyset$ , et la meilleure solution  $S^*$ .

**Etape 3 :** les agents de planification se déplacent le long des projets triés pour construire des solutions en sélectionnant par probabilité un certain nombre d'unités de chaque projet (Eq. 8). En exploitant le budget restant et en utilisant la méthode gloutonne (*greedy simple*), chaque plan résultant est amélioré, puis ajouté à l'ensemble de solutions  $S$ .

**Etape 4 :** met à jour la meilleure solution  $S^*$  si une solution de  $S$  est meilleure que celle-ci, puis modifie les informations phéromoniques en prenant en considération des solutions générées au cours de la dernière itération (Eq. 3).

**Etape 5 :** si la condition d'arrêt n'est pas vérifiée, alors revenez à l'étape 3.

**Etape 6 :** retourne le meilleur plan urbain  $S^*$  et ses caractéristiques.

A l'étape 5, la condition d'arrêt est atteinte lorsque la fonction objectif stagne. Pour obtenir des plans matures, un minimum d'itérations ( $MinIter$ ) doit être réalisé. Pour éviter une boucle infinie, l'algorithme se termine inévitablement à  $MaxIter$ .

### 3.10.5. L'algorithme H-MACO

L'algorithme d'optimisation multi-objectifs par colonies de fourmis du système holonique (H-MACO) applique un processus d'optimisation ascendant en appelant récursivement pour chaque holon sous-zone l'algorithme de colonies de fourmis modifié (BKPACS) qui applique ses propres informations phéromonique et heuristique.

Pour un plan d'une meilleure adéquation globale des projets urbains  $S^*$ , les agents de planification interagissent dans les trois niveaux d'holons de planification des projets urbains et se coordonnent simultanément en vue d'obtenir une solution pseudo-optimale

vis-à-vis les principaux objectifs (préférence, coût, temps et qualité). Les étapes suivantes expliquent clairement le fonctionnement de l'algorithme H-MACO :

### **Etape 1 : Initialisation**

- a) En tant qu'holon supérieur de la holararchie  $H$ , l'holon de territoire urbain s'adresse vers le bas et regroupe ses holons de zone pour évaluer les ressources nécessaires. Tandis que chaque holon de base  $h_i$  définit ses besoins  $N_i = (u_i^1, \dots, u_i^{m_i})$  à son super-holon.
- b) L'holon de territoire met en place la stratégie de développement du projet urbain, et alloue le budget à chaque holon de zone, qui à son tour répartit ce budget à ses holons subordonnés (holons des sous-zones) et fixe ainsi ses objectifs.
- c) Charger le catalogue de projets urbains qui contient toutes les informations de chaque holon de base (holon de sous-zone) : les projets  $p_i$ , les préférences des projets  $v_i$  et les coûts  $c_i$ .
- d) Initialiser les pondérations des préférences globales  $w_i$  à 1 pour chaque projet  $p_i$ , où  $w_i$  est un réel  $\in ]0,1]$  utilisé pour réguler le nombre d'unités à sélectionner de chaque projet au niveau le plus bas (niveau de planification locale) afin d'être le plus proche des objectifs globaux du plus haut niveau (niveau de planification globale) sans dépasser les limites globales.

**Etape 2 : Optimisation ascendante :** exploration de l'holarchie depuis les parties vers le tout :

- a) Connaître les valeurs  $(c_i, v_i, u_i, B)$  de chaque projet  $p_i$  à développer pour chaque sous-zone urbaine dans l'holon du niveau le plus bas permet de démarrer le processus d'optimisation. Où  $c_i$  est le coût unitaire du projet  $p_i$ ,  $v_i$  est la valeur de préférence,  $u_i$  est la borne supérieure des unités de  $p_i$  et  $B$  représente le budget alloué.
- b) En appliquant l'algorithme BKPACS, les agents (fourmis) de chaque holon de base agissent pour trouver les plans optimaux locaux, puis les communiquent au niveau supérieur sous forme d'agrégats de plans locaux.
- c) Les agents de planification du niveau d'agrégation œuvrent pour sélectionner la meilleure combinaison des plans agrégés qui seront consolidés dans le niveau supérieur. En respectant l'échelle globale, l'opération de consolidation de plans modifie la structure de l'unité de projet de manière à permettre la vérification et l'optimisation des plans à l'échelle globale.

**Etape 3 : Amélioration descendante des plans**

- a) Une fois reçus les plans consolidés, l'holon supérieur évalue leur optimalité par rapport aux objectifs globaux, puis communique les directives à ses subordonnés pour éventuelles négociation et amélioration du plan.
- b) Chaque holon de zone ajuste progressivement les valeurs de pondérations  $w_i$  pour tous les projets faisant partie des agrégations qui divergent des objectifs globaux.

- c) Mettre à jour les préférences des projets en appliquant les valeurs de pondération globales des projets, de sorte que la préférence  $v_i$  prenne la nouvelle valeur :  $v_i \cdot w_i$ .
- d) Selon le résultat de l'évaluation, le processus d'optimisation itère à partir de l'étape 2 jusqu'à l'optimisation locale et globale des plans ou atteindre le nombre d'itérations maximum.

Le pseudo-code suivant (Algorithm 1) résume le fonctionnement cet algorithme :

---

**Algorithm 1:** H-MACO algorithm

---

**Inputs:** urban projects' catalog  $C$ , urban projects holarchy  $H$ , urban needs  $N$   
**Outputs:** the best global urban plan  $S^*$   
**Initialization:** Global objectives  $O$ , allocate budget  $B$ , initialize global preference weights  $W$   
 $S^* \leftarrow \emptyset$  {Global urban plan}  
**Repeat** {Level 2}  
     $S_1 \leftarrow \emptyset$  {First level urban plans}  
    **For each**  $Area \in Area\_Holons$  **do** {Level 1}  
         $S_0 \leftarrow \emptyset$  {Local urban plans}  
        **For each**  $Sub-area \in Sub-area\_Holons(Area)$  **do** {Level 0}  
             $S_0 \leftarrow S_0 + BKPACS(Sub-area)$  {Aggregate optimal local solutions}  
        **End For**  
         $S_1 \leftarrow S_1 \cup S_0$   
    **End For**  
     $S^* \leftarrow FindeBest(S_1)$   
    Update ( $W$ )  
**Until** *terminationConditionSatisfied*  
**return**  $S^*$

---

Ces algorithmes demeurent utiles aux agents holoniques du système de la planification des projets urbains dans leur fonctionnement et leur interaction. Il en découle que chaque agent doit connaître les compétences de ses partenaires en termes de planification des projets urbains. Ainsi, ils s'adaptent et coopèrent avec d'autres agents appartenant à des systèmes plus grands pour satisfaire des objectifs communs.

Dans la section suivante nous présentons le fonctionnement de notre système intelligent d'aide à la décision conçue pour la planification des projets urbains à travers une description explicite de ses sous-systèmes.

### 3.11. Description et fonctionnement du SIAD-PPU

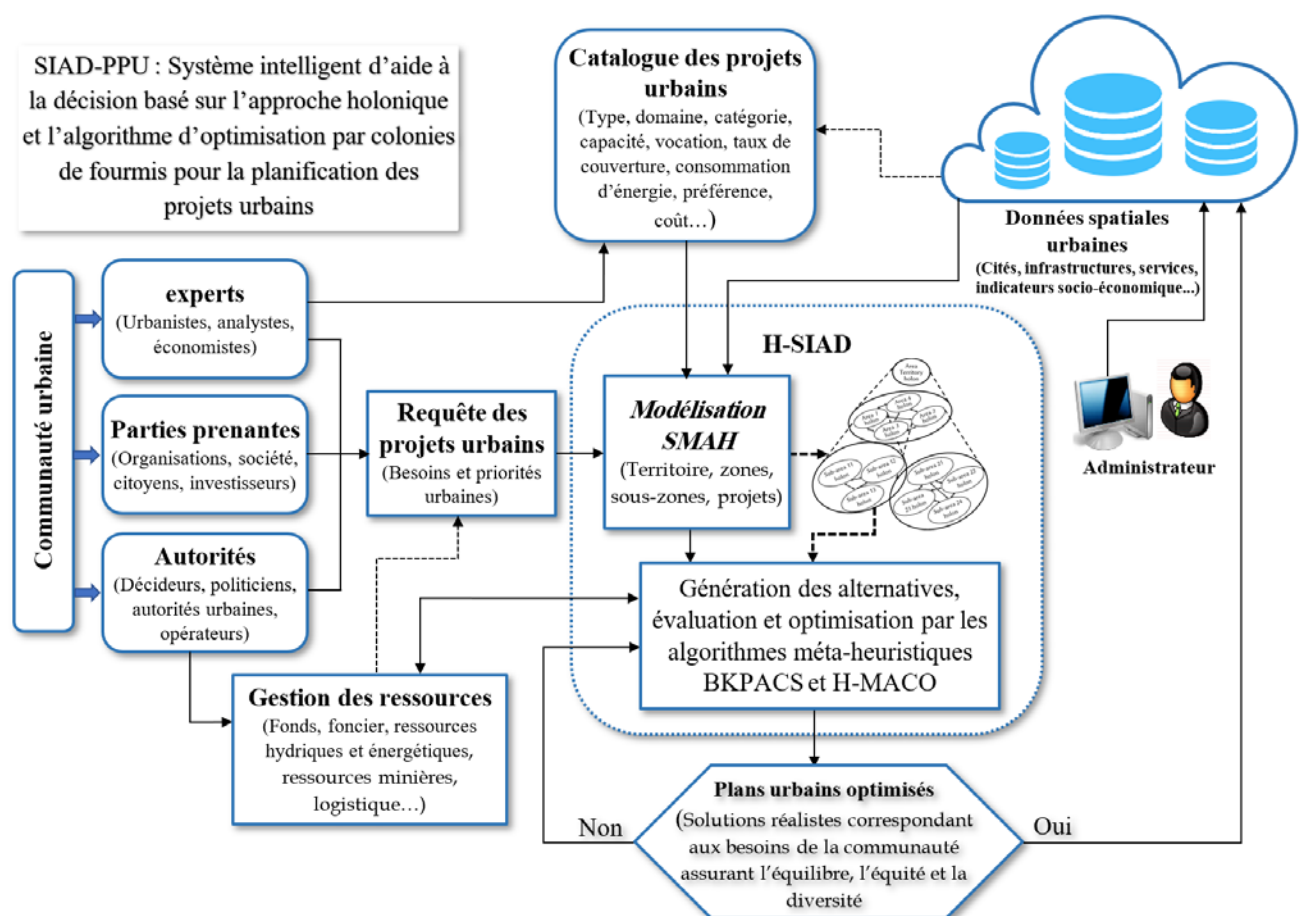
Un système d'assistance aux acteurs de la communauté urbaine (experts urbains, économistes, décideurs, citoyens...) est d'une importance majeure dans la réussite de la mise en place des projets urbains.

Afin d'augmenter à la fois l'autonomie des acteurs dans la gestion des zones urbaines et la coopération par la prise en compte des actions communes sur le territoire urbain, ce système d'aide à la décision doit être construit selon les besoins et les attentes de l'ensemble acteurs en vue de produire des plans des projets urbains permettant à la ville de bien paraître, de bien fonctionner, et de bien évoluer.

Les procédures de la planification urbaine se basent essentiellement sur l'analyse du territoire urbain et sur le diagnostic de son état actuel. Ainsi l'élaboration de meilleurs plans nécessitent d'introduire des techniques intelligentes qui facilitent la projection à l'échelle temporelle des opérations urbaines à concrétiser sur le territoire.

D'autant plus, pour garantir une bonne performance du système, il est indispensable d'accorder une attention particulière à l'organisation hiérarchique des sous-systèmes en coordination dont les tâches sont bien définies.

L'objectif est en effet d'obtenir une organisation holonique et intelligente, capable de modéliser le système de la planification des projets urbains dans un environnement cohérent (Figure 3.13.), offrant à l'ensemble des utilisateurs les outils nécessaires. Ce système comprend principalement les fonctions suivantes :



**Figure 3.13.** Système intelligent d'aide à la décision basée sur l'approche holonique pour la planification des projets urbains (Source : [Khelifa and Laouar (2020)])

- a) **Acquisition des données** : c'est d'acquérir les données concernant les zones urbaines il s'agit des données spatiales, infrastructures, indicateurs socio-économiques, etc.
- b) **Mise à jour des catalogues et référentielles urbains** : les zones urbaines pouvant accueillir des projets urbains selon les besoins exprimés par la communauté urbaine. Les

caractéristiques des projets, telles que l'unité élémentaire, les préférences et les coûts sont décrits dans un catalogue spécifique qui doit être mis à jour régulièrement.

**c) Gestion des ressources urbaines :** en collaboration avec l'ensemble des acteurs, les décideurs ont le pouvoir de gérer les différentes ressources (les fonds, le foncier, et la logistique...) afin de garantir une exploitation rationnelle et de prévoir de meilleurs plans urbains.

**d) Définir les besoins et priorités urbaines :** cette tâche est confiée à la communauté urbaine notamment les décideurs (autorités locales, politiciens), les experts (urbanistes, architectes, analystes, économistes, ...), et les parties prenantes (citoyens, associations, investisseurs, ...), elle consiste à exprimer les besoins et les priorités en matière de projets urbains.

**e) Planification des projets urbains :** elle constitue le sous-système holonique intelligent d'aide à la décision (H-SIAD) chargé de traiter les requêtes provenant de la communauté urbaine. En se référant au catalogue, et en utilisant les données spatiales, les besoins en projets urbains sont intégrés dans un modèle holonique respectant la hiérarchie du territoire urbain structuré en zones, sous-zones, ...

Les holons formant notre sous-système coopèrent ensemble en exploitant les ressources allouées, afin de générer des variantes de plans de projets urbains et d'en choisir les meilleurs, répondant aux besoins locaux et réalisant les objectifs souhaités par le système global. L'optimisation et l'évaluation des plans des projets urbains y s'effectuent par l'application des algorithmes BKPACS et H-MACO.

**f) Mise à jour des bases de données géo-spatiale :** les plans urbains optimaux résultants sont en fait des solutions réalistes, qui répondent aux besoins de la communauté urbaine de façon équitable, équilibrée, et diversifiée. Ces plans sont soumis à être approuvé par les décideurs, puis intégrés dans la base de données spatiale pour une éventuelle mise à jour.

### **3.12. Conclusion**

Les SMAH sont très connus par leurs applications dans divers domaines. En plus des propriétés communes héritées des SMA, ils se distinguent principalement par la capacité d'agent holonique d'être à la fois un composant d'un élément de niveau supérieur, et un tout composé d'autres agents dans une organisation hiérarchique récursive dite holarchie.

Les SMAH semblent être une solution prometteuse pour concevoir des systèmes intelligents et coopératifs, et qui fournissent une alternative intéressante pour la planification urbaine. En effet, l'idée revient à tirer profit des SMAH, afin de développer un système intelligent d'aide à la décision pour la planification des projets urbains, capable de répondre aux exigences informationnelles, organisationnelles, et fonctionnelles à travers une architecture holonique.

Dans ce chapitre nous avons introduit l'approche holonique, considérée comme un outil puissant de modélisation des systèmes distribués, complexes et à grande échelle. En se


basant sur l'agent holonique comme étant le noyau des systèmes multi-agents holoniques, sur lesquels nous avons fondé notre travail.

Nos contributions couvrent principalement la conception et la modélisation de la planification des projets urbains fondée sur l'application l'approche holonique. D'ailleurs, le défi majeur était d'employer les techniques intelligentes d'aide à la décision notamment les méthodes méta-heuristiques et de les adapter aux agents holoniques, en vue de renforcer le processus de la planification des projets urbains tant sur le niveau local que global.

Pour ce faire, les agents holoniques appliquent l'algorithme BKACS pour assurer l'optimisation locale des plans des projets urbains considérée comme étant un problème du sac-à-dos borné (*bounded knapsack problem* BKP), ainsi que l'algorithme H-MACO qui coordonne entre les holons des différents niveaux afin de réaliser une optimisation globale du système.

En fin de chapitre, une description détaillée du fonctionnement du système proposé ainsi que les algorithmes d'optimisation développés (BKPACS, H-MACO), dont les expérimentations seront exposées dans le chapitre suivant.



 *Chapitre-4*

# Expérimentation & Evaluation



## Chapitre - 4

# EXPERIMENTATION ET EVALUATION

### 4.1. Introduction

Dans ce chapitre nous allons passer à l'analyse et à l'évaluation de notre système intelligent d'aide à la décision destiné à la planification des projets urbains, par l'implémentation d'un prototype permettant d'effectuer des expérimentations sur des cas d'étude et des simulations pour tester l'efficacité et la performance des algorithmes BKPACS et H-MACO développés au sein de cette thèse.

Afin de voir leurs apports par rapport à d'autres algorithmes comme les algorithmes génétiques, nous avons établi d'une part une comparaison des résultats obtenus durant le processus l'optimisation locale et d'autre part une évaluation des résultats du processus d'optimisation globale.

### 4.2. Le prototype SIAD-PPU

Avant d'entamer les expérimentations, nous allons présenter brièvement notre prototype (Figure 4.1.) du système intelligent d'aide à la décision de la planification des projets urbains (SIAD-PPU), dont les principales fonctions sont les suivantes : l'acquisition des informations concernant les zones urbaines et infrastructures, le catalogue des projets urbains ; la formulation des requêtes (exprimées en besoins de projets urbains par zone) ; définir les objectifs et la gestion des ressources ; l'initialisation des paramètres nécessaires pour exécuter les algorithmes d'optimisation des plans des projets urbains et à la structure des agents holoniques.

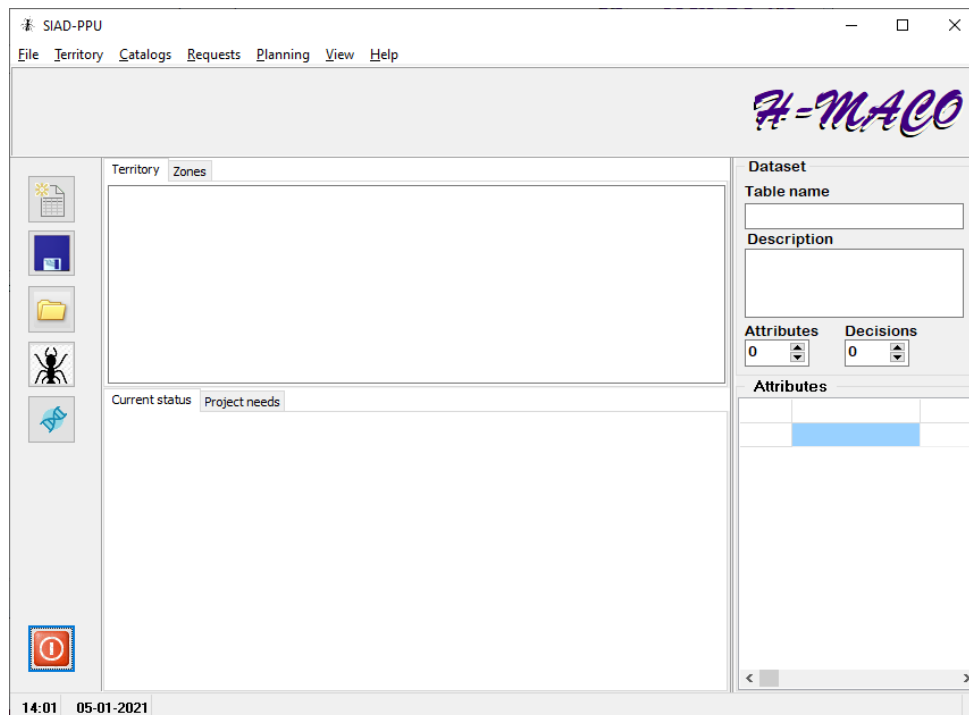


Figure 4.1. Interface principale du prototype SIAD-PPU

L'interface principale de l'application permet à l'utilisateur de mettre à jour les données propres à chaque zone urbaine y compris le catalogue local des projets urbains. Au fur et à mesure l'organisation holonique se développe en donnant à l'utilisateur (qui représente un agent holonique) sa position définie à la fois par son supérieur et ses subordonnés dans la hiérarchie à partir de laquelle ils tracent leurs objectifs.

#### ***4.2.1. Administration des données***

D'après sa nature, le système de la planification des projets urbains est un système distribué qui manipule des informations et des données appartenant à plusieurs secteurs d'activités et réparties sur tout un territoire. Ces données sont mises à la disposition aux administrateurs désignés ayant les droits de manipuler, de gérer, et de partager ces données.

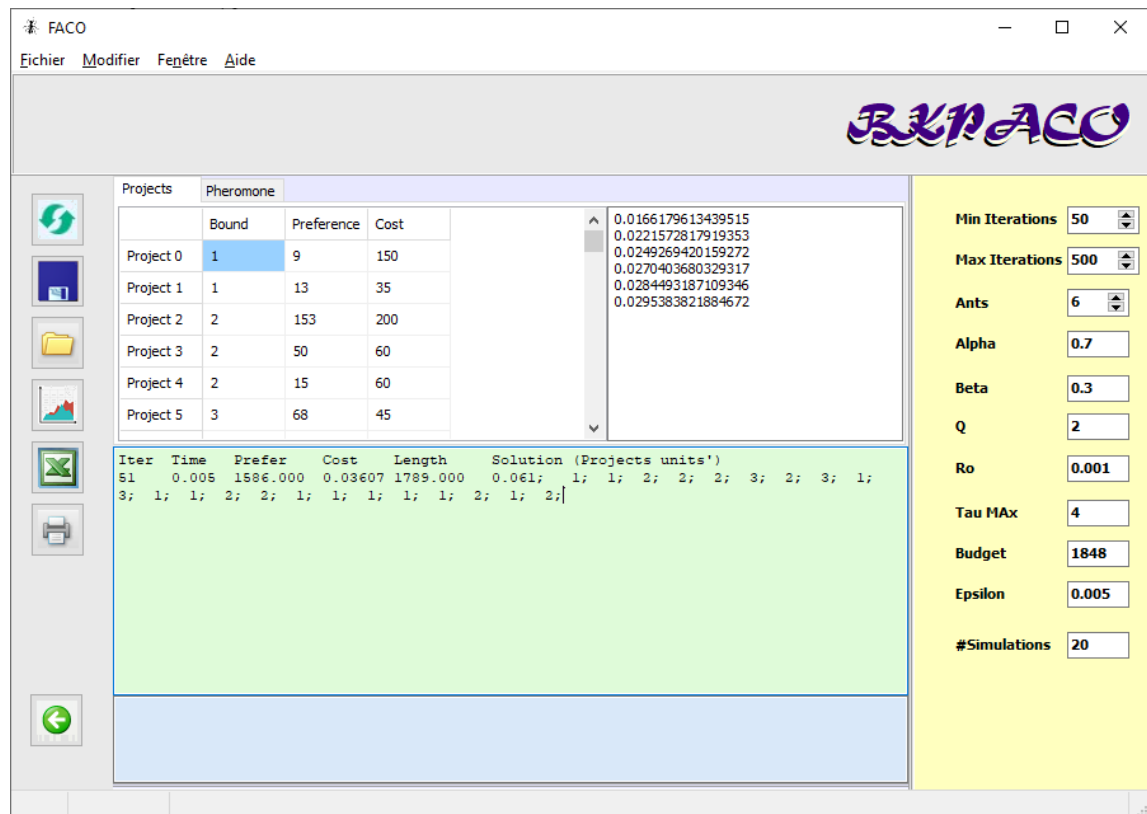
Les données relatives à la planification urbaine comprennent essentiellement les détails de chaque zone urbaine (infrastructures, coordonnées, adjacence, indicateurs démographique et socio-économique, vocation principale, ressources, etc.), le catalogue des projets urbains (type de projet, domaine, catégorie, capacité d'accueil, taux de couverture, consommation énergétique, préférence, coût), impact socio-économique et environnemental, etc.

L'échange de ces informations se fait par internet en utilisant des méthodes standardisées comme *eXtensible Markup Language* (XML) et *Javascript Object Notation* (JSON).

#### ***4.2.2. Initialisation des paramètres***

Les algorithmes d'optimisation des plans des projets urbains nécessitent essentiellement l'initialisation de paramètres d'exécution et d'ajustement des entrées et des sorties, à titre d'exemple, l'algorithme BKPACS utilise les paramètres suivants (Figure 4.2) :

- *Min iterations* : c'est le nombre minimum d'itérations lors d'un cycle d'optimisation pour assurer des plans matures.
- *Max iterations* : c'est le nombre maximum d'itérations pour éviter une boucle infinie.
- *Ants* : c'est le nombre d'agents artificiels (fourmis) dans chaque holon de base.
- *Alpha* et *Beta* : ce sont des paramètres pour ajuster respectivement, la quantité de la phéromone et la valeur heuristique dans la probabilité de sélection d'unités de projets.
- *Q* : c'est un paramètre pour normaliser les valeurs de la quantité de la phéromone et les rendre significatives.
- *Ro* : c'est le taux d'évaporation de la phéromone.
- *Tau Max* : utile pour initialiser l'information phéromonique.
- *Budget* : le montant du budget alloué.
- *Epsilon* : utile pour la condition d'arrêt.
- *#Simulation* : nombre d'exécutions lors d'une simulation.



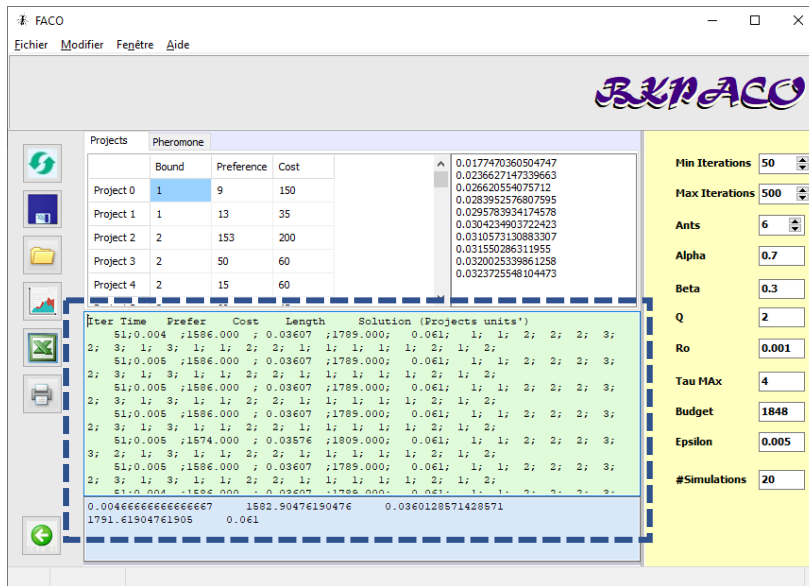
**Figure 4.2.** Initialisation des paramètres et déroulement d'un exemple d'optimisation par l'algorithme BKPACS

### 4.2.3. La formulation des requêtes

Cette application permet de formuler simultanément des requêtes utilisateurs, ayant en entrées les besoins exprimés en unités de projets urbains et préférences et en se servant du catalogue des projets urbains, et bien entendu dans la limite des ressources allouées, cette application retourne des plans urbains optimaux (Figure 4.2.).

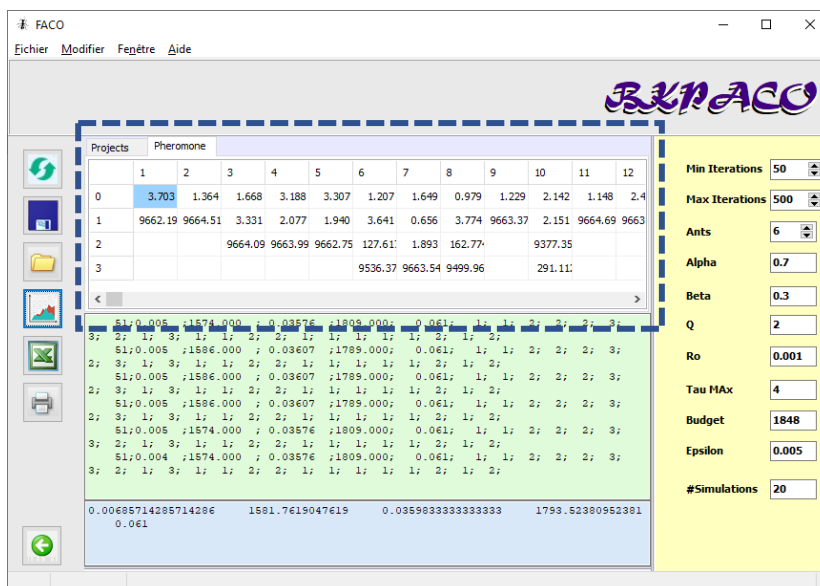
### 4.2.4. L'optimisation des plans des projets urbains

Comme nous l'avons déjà mentionné dans le chapitre 3, la planification des projets urbains est considérée comme étant un problème d'optimisation du sac à dos, où les requêtes cherchent les meilleures combinaisons des projets urbains qui maximisent la fonction objectif construite autour des préférences des projets sous la contrainte des ressources allouées (budget, qualité, temps, etc.). L'algorithme BKPACS fournit des plans urbains optimaux au niveau local et en un temps réduit par rapport à la dimension du problème défini par le nombre de projets et le nombre d'unités par projet. Pour cela des simulations ont été effectuées, comme illustré dans la figure (Figure 4.3.)



**Figure 4.3.** Simulation d'optimisation des projets urbains par l'algorithme BKPACS

L'application permet de visualiser l'évolution de la phéromone lors du processus d'optimisation, qui d'une importance majeure dans la phase d'expérimentation et d'évaluation de cette application (Figure 4.4.).



**Figure 4.4.** L'information phéromonique lors d'un processus d'optimisation par l'algorithme BKPACS

Nous avons également intégré et adapté l'algorithme génétique d'optimisation pour la raison d'une comparaison montrant l'efficacité et la performance de notre algorithme BKPACS (Figure 4.5.).

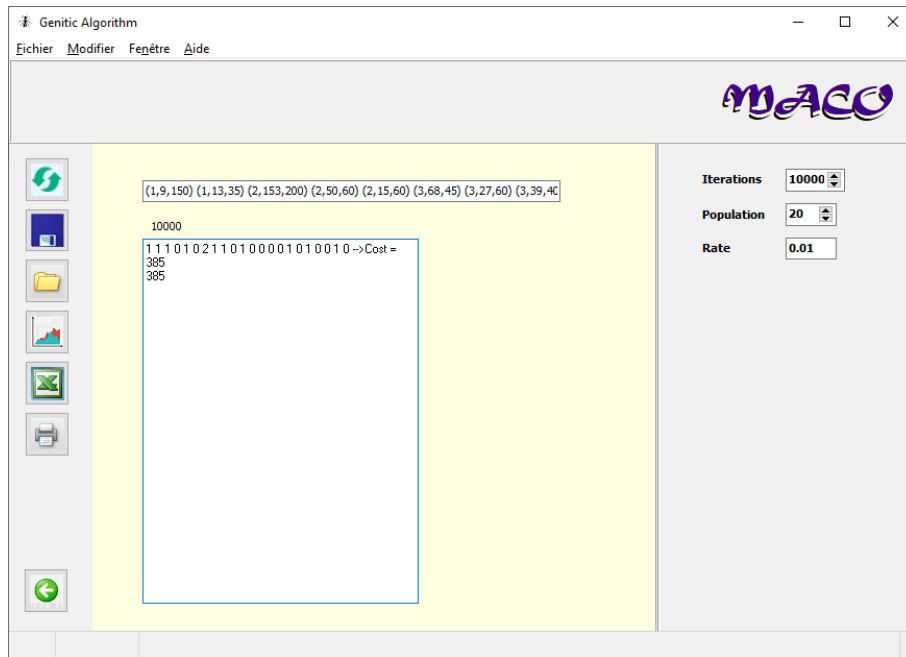


Figure 4.5. Optimisation des plans des projets urbains par l'algorithme génétique

### 4.3. Apports et limites du SIAD-PPU

Actuellement, notre SIAD-PPU est en phase d'expérimentation, tandis que nos préoccupations principales étaient le développement et l'amélioration des algorithmes de base notamment BKPACS et H-MACO nécessaires à l'optimisation des plans des projets urbains. En s'inspirant des algorithmes méta-heuristiques nous avons adapté l'algorithme d'optimisation par colonies de fourmis pour résoudre des problèmes combinatoires, multi-objectifs et à grande dimension en un temps minimal.

En fait, la complexité du problème de la planification des projets urbains nous a imposé de réorganiser notre système en décomposant le territoire urbain en zones et en sous zones urbaines, et d'admettre qu'un projet urbain est formé d'unités élémentaires à réaliser pour satisfaire les besoins de la communauté urbaine. C'est pour cela recourir à l'approche holonique demeure une nécessité, d'après ce qu'elle met à notre disposition en matière de moyens d'organisation et d'outils de modélisation des systèmes complexes, distribués et à grande échelle.

Pour valoriser les apports concrétisés dans notre système intelligent d'aide à la décision, plusieurs expérimentations ont été menées sur les composants principaux notamment l'algorithme méta-heuristique d'optimisation multi-objectifs BKPACS, développé pour construire des solutions pseudo-optimales, et l'algorithme H-MACO qui assure l'intégration de l'ensemble des holons constituant la holarchie du système urbain.

#### 4.3.1. Expérimentations et discussion des résultats

Le système proposé sera testé sur des cas d'étude extraits de l'étendu urbain de la wilaya de Tébessa, située à l'est de l'Algérie, qui s'étale sur une superficie totale de 5358,32 milles

carrés, divisée en 12 daïras, qui sont à leur tours divisés en 28 communes avec une population globale de plus de 768 188 (Estimation : 2018).

En superposant l'organisation hiérarchique de la wilaya de Tébessa à la structure holonique, les 12 daïras (zones) forment le premier niveau de subdivisions administratives alors que les communes (sous-zones) forment le second niveau.

L'exemple suivant illustre l'application de l'algorithme BKPACS sur une sous-zone pour planifier un ensemble de projets avec un budget de 450 000 \$. Les informations utiles sont résumées dans le tableau 4.1, la 4<sup>ème</sup> colonne donne le nombre actuel d'unités de chaque projet, et les besoins sont exprimés dans la colonne 5. La 6<sup>ème</sup> colonne appelée plan BKP représente une solution par la méthode gloutonne (*greedy simple*).

Projet	Préférence	Coût \$(10 <sup>3</sup> )	Courant	Besoins	Plan BKP
$P_1$	70	3333	10	4	<b>4</b>
$P_2$	60	4167	16	6	<b>6</b>
$P_3$	55	5000	50	3	<b>3</b>
$P_4$	25	8333	80	5	<b>5</b>
$P_5$	20	12500	30	9	<b>5</b>
$P_6$	90	22500	25	7	<b>7</b>
$P_7$	60	26667	15	6	<b>5</b>
$P_8$	50	125000	45	9	<b>0</b>

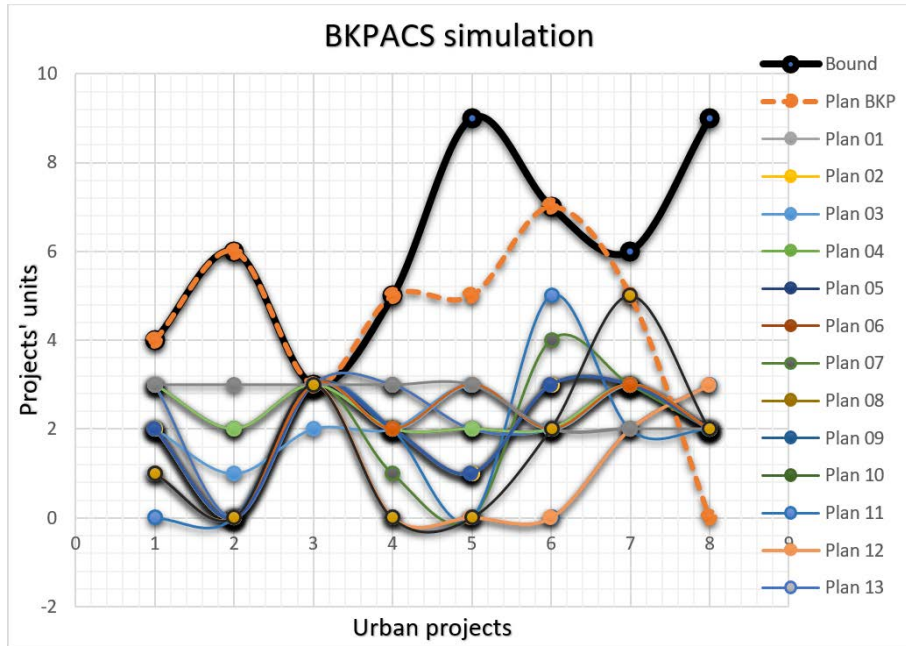
**Tableau 4.1** - les données des projets d'une zone urbaine

Pour chaque commune, les besoins du projet sont exprimés en nombre d'unités (Tableau 4.1), chaque unité élémentaire a ses propres propriétés (préférence et coût) collectées à partir du catalogue d'unités élémentaires du projet (Tableau 4.2).

ID	Projet	Unité	Préférence	Coût (\$)
$P_1$	Ecole primaire	Classe	70	3333
$P_2$	Ecole moyenne	Classe	60	4167
$P_3$	Lycée	Classe	55	5000
$P_4$	Cantine scolaire	Cantine	25	8333
$P_5$	Logement (F2)	Appartement	20	12500
$P_6$	Logement (F3)	Appartement	90	22500
$P_7$	Logement (F4)	Appartement	60	26667
$P_8$	Admin. Infrastructure	Unité	50	125000
...				

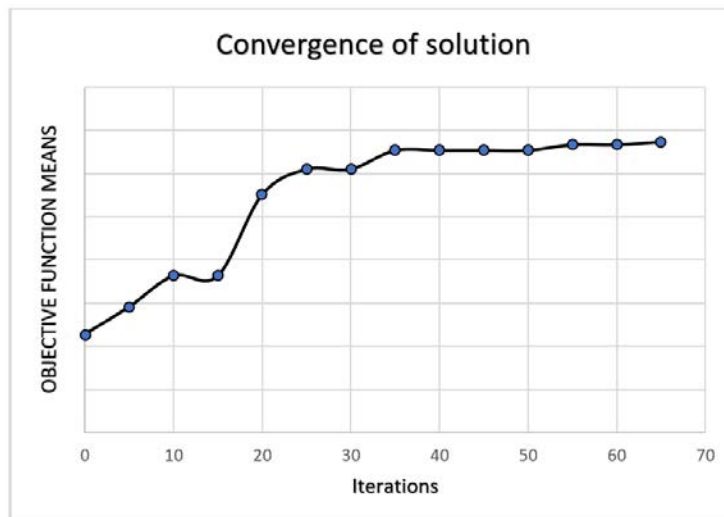
**Tableau 4.2** – Catalogue des projets urbains

En appliquant l'algorithme BKPACS, les plans générés sont illustrés dans la figure 4.6 :



**Figure 4.6.** : Une variété de plans des projets urbains générés par l'algorithme BKPACS comparés à celui généré par l'algorithme *greedy simple*.

Les plans qui en résultent par la méthode gloutonne (*greedy simple*) rejoignent la borne des projets d'une efficacité ( $e_i = v_i/c_i$ ) élevée ; cependant, les projets à faible efficacité ne sont pas du tout considérés, à cause de l'épuisement précoce du budget. Lors de la construction des plans optimaux, l'algorithme BKPACS apporte des solutions plus réalistes tout en respectant les critères d'équilibre et de diversité dans le choix des projets à promouvoir. Cet algorithme assure la convergence des solutions optimales dans un temps raisonnable (Figure 4.7.).



**Figure 4.7.** Convergence des solutions par l'algorithme BKPACS

Pour évaluer l'efficacité et la performance de l'algorithme BKPACS, nous avons utilisé trois cas d'étude chacun d'entre eux concerne une zone urbaine dont la structure des



données est similaire à celle de l'exemple précédent (tableau 4.1). Plus de 2000 simulations ont été appliquées respectivement sur 25%, 50%, 75% de la totalité des budgets nécessaires pour couvrir tous les projets de chaque cas d'étude, et en se concentrant sur les caractéristiques des plans optimaux, les résultats générés par l'algorithme BKPACS exprimés en valeurs moyennes sont meilleurs par rapport à ceux obtenus en appliquant l'algorithme génétique (Cf. tableau 4.3).

<b>Cas d'étude N°1 budget alloué (1.650.001,00 \$), nombre de projets 08</b>				
<b>Méthode</b>	<b>Budget %</b>	<b>Temps machine (ms)</b>	<b>Efficacité <math>\sum u.v/c</math></b>	<b>Exploitation du budget</b>
<i>Algorithme BKPACS</i>	25%	4.28	0.66	88.14 %
	50%	5.72	0.84	92.32 %
	75%	6.85	0.92	98.12 %
<i>Algorithme génétique</i>	25%	1.28	0.50	80.47 %
	50%	4.68	0.75	95.16 %
	75%	8.82	0.89	83.41 %

<b>Cas d'étude N°2 budget alloué (11.000.000,00 \$), nombre de projets 50</b>				
<b>Méthode</b>	<b>Budget %</b>	<b>Temps machine (ms)</b>	<b>Efficacité <math>\sum u.v/c</math></b>	<b>Exploitation du budget</b>
<i>Algorithme BKPACS</i>	25%	18.02	0.63	90.11 %
	50%	25.16	0.85	95.27 %
	75%	24.31	0.97	94.78 %
<i>Algorithme génétique</i>	25%	16.88	0.56	90.17 %
	50%	30.47	0.70	89.10 %
	75%	52.49	0.59	91.50 %

<b>Cas d'étude N°3 budget alloué (100.000,000,00 \$), nombre de projets 100</b>				
<b>Méthode</b>	<b>Budget %</b>	<b>Temps machine (ms)</b>	<b>Efficacité <math>\sum u.v/c</math></b>	<b>Exploitation du budget</b>
<i>Algorithme BKPACS</i>	25%	60.01	0.64	98.14 %
	50%	72.55	0.76	95.54 %
	75%	73.63	0.95	97.12 %
<i>Algorithme génétique</i>	25%	22.54	0.72	79.13 %
	50%	80.72	0.88	82.77 %
	75%	150.35	0.62	79.31 %

**Table 4.3.** Comparaison des plans pseudo-optimaux des projets urbains générés par l'algorithme BKPACS par rapport à l'algorithme génétique.

D'après les résultats expérimentaux relatifs aux cas d'étude, l'algorithme BKPACS se montre compétitif en produisant des plans urbains optimaux au niveau local en un temps relativement réduit, aussi bien dans des situations présentant un espace d'états considérable. Il permet également de satisfaire des fonction multi-objectifs et d'introduire de nouvelles contraintes voir même des contraintes non-linéaires.

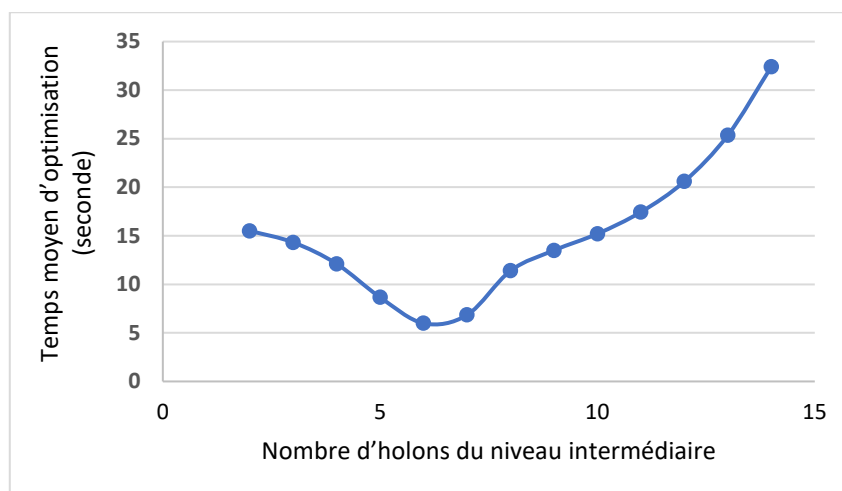
En ce qui concerne l'algorithme H-MACO, notre cas d'étude comprend autant d'holons de base que de zones urbaines représentant les 28 communes de la wilaya de Tébessa. Prenons 20 types de projets urbains à promouvoir, dont les besoins de chaque zone ne dépassent pas 100 unités de chaque projet, il s'agit d'étudier le comportement de cet

algorithme par rapport au nombre d'holons formant le niveau intermédiaire. Nous allons regrouper les holons de base deux à deux, puis trois à trois, et ainsi de suite, et en calculant au fur à mesure le temps d'optimisation global. Les résultats de cette expérimentation sont récapitulés dans le tableau 4.4.

Nombre d'holons du niveau intermédiaire	Temps moyen d'optimisation (seconde)
2	15.51
3	14.32
4	12.11
5	8.66
6	5.99
7	6.85
8	11.40
9	13.50
10	15.21
11	17.45
12	20.60
13	25.36
14	32.41

**Tableau 4.4.** Evaluation du temps moyen d'optimisation global des plans des projets urbains par l'algorithme H-MACO

En fixant le nombre d'holons de base du territoire urbain à étudier, et en changeant à chaque expérience ceux du niveau intermédiaire, on remarque que le temps moyen d'optimisation diminue jusqu'à la valeur 5.99 secondes, ce qui correspond à 6 holons, ensuite les valeurs du temps moyen d'optimisation s'accroissent progressivement à chaque fois qu'on augmente le nombre d'holons du niveau intermédiaire.



**Figure 4.8.** Variation du temps moyen d'optimisation global par l'algorithme H-MACO

Le graphe ci-dessus (Figure 4.8) illustre la variation du temps d'optimisation global en fonction du nombre d'holons du niveau intermédiaire par l'algorithme H-MACO. D'après cette expérimentation, on constate que le nombre d'holons du niveau intermédiaire influence remarquablement sur le temps d'optimisation, où le meilleur temps est obtenu lorsque le nombre d'holons dans le niveau intermédiaire (niveau d'agrégation) voisine les 20% du nombre d'holons de base ; cependant plus le nombre d'holons du niveau intermédiaire s'éloigne de la valeur 20%, plus le temps d'optimisation augmente relativement.

Le renforcement du niveau intermédiaire par des holons permet de décomposer le problème d'optimisation davantage, en contrepartie il augmente le trafic de communication causés par l'interconnexion des holons des différents niveaux. Il faudrait donc construire soigneusement la structure de l'holarchie afin d'accélérer le processus d'optimisation en choisissant les nombre d'holons convenable à chaque niveau.

Considérons le nombre des zones urbaines égal à 28, nous avons appliqué l'algorithme H-MACO pour optimiser les plans urbains de différentes dimensions (exprimés en nombre projets par zone). Le nombre d'unités de chaque projet à affecter à une zone ne doit pas dépasser 100 unités avec un budget équivalent à 50% des besoins exprimés par chaque zone. Les résultats obtenus sont inscrits dans le tableau suivant (Tableau 4.5) :

Expérience	Dimension du problème (Nb projets/zone)	Plan optimal (Itérations)	Exploitation du budget %	Efficacité (Préférences/coût)	Temps d'optimisation
1	25	5350	95.25%	0.75	7.23 sec
2	50	9534	96.48%	0.82	10.34 sec
3	100	12000	97.42%	0.64	15.27 sec
4	200	13000	93.21%	0.79	19.31 sec
5	400	15000	96.57%	0.87	25.54 sec

**Tableau 4.5** – Résultat récapitulatif des expériences sur l'algorithme H-MACO

A titre d'exemple, la première expérience a été menée sur les 28 zones urbaines pour planifier 25 projets urbains, chaque projet est représenté au maximum par 100 unités dont l'optimisation globale s'est déroulée pendant 5350 itérations (calculées à partir des itérations d'optimisation locale faites par l'algorithme BKPACS dans chaque holon). Par conséquent, le taux d'exploitation du budget est de 95.25%, l'efficacité du plan obtenu évaluée par le rapport préférences/coût est estimée et normalisée dans l'intervalle  $[0,1]$  à 0.75, et finalement le temps consommé par le processus d'optimisation est de l'ordre de 7.23 secondes.

En comparant les cinq expériences, on conclut que le passage à l'échelle semble être satisfaisant de point de vue optimisation réalisant les objectifs un temps limité. En fait, le taux d'exploitation budgétaire est d'une moyenne de 95.79%, la moyenne de l'efficacité des plans est 0.77 se rapproche de la borne supérieure, le temps d'optimisation suit une fonction linéaire.

### **4.3.2. Limites du SIAD-PPU**

Bien que notre système ait prouvé son efficacité d'optimiser les plans des projets urbains, il demeure en besoin d'être appliqué dans diverses situations du monde réel, car les tests sur des données variées garantissent la performance et la fiabilité de notre système, et spécialement en l'absence d'un outil similaire ou d'une plateforme de test et de validation.

Il est à noter que le paramétrage des algorithmes d'optimisation n'a pas eu sa part dans cette thèse, parce que nous avons concentré plus précisément sur la faisabilité de la méthodologie employée et son utilité pour résoudre le problème de la planification des projets urbains, que nous avons trouvé insoluble lorsqu'il atteint une dimension importante.

D'ailleurs le prototype développé s'avère limité et ne dispose que des modules essentiels pour le test du système, ce qui nécessite une amélioration profonde par l'intégration de nouveaux modules tels que : gestion avancée des catalogues des projets urbains, exploitation des données géo-spatiales provenant des APIs SIG et les sources des données ouvertes (Natural Earth Data , Esri Open Data , Open-StreetMap , OpenStreetMap , and FAO GeoNetwork4), ainsi que la visualisation de la structure holarchique permettant une interaction conviviale avec le système holonique de la planification des projets urbains.

Dans la pratique, et pour être plus réaliste, aucune approche n'est parfaite pour couvrir tous les enjeux du domaine de la recherche, selon *Binmore* [Binmore (1990)] aucun algorithme de calcul n'est à la fois complet et rationnel. Tant que la rationalité dans toute situation étant relativement impossible, il suffit alors d'employer les moyens disponibles offrant de meilleurs avantages. En fait, l'algorithme BKPACS développé répond de manière flexible au sein du système holonique, qui à son tour fonctionne efficacement pour la modélisation des systèmes distribués à grande échelle.

### **4.4. Conclusion**

Dans ce chapitre nous avons essayé d'évaluer nos contributions, à travers l'application des algorithmes développés suivant des expérimentations variées. Pour cette raison nous avons exploité notre prototype SIAD-PPU, qui nous a permis de simuler le système holonique multi-agents de la planification des projets urbains et d'en dégager les résultats de la performance et l'efficacité des algorithmes BKPACS et H-MACO qui se sont montrés compétitifs et en particulier lorsqu'il s'agit d'un problème de planification à grande échelle, dont l'objectif est de réaliser à la fois une optimisation locale et globale.

Exploiter des algorithmes de colonies de fourmis est un acquis ouvrant la possibilité de les appliquer pleinement dans différents domaines, car leur potentiel pourrait apporter une meilleure solution d'optimisation aux problèmes complexes de la vie. En effet, les résultats obtenus des algorithmes de colonies de fourmis au sein d'un système holonique ont donné des résultats performants par rapport à ceux d'autres méthodes.

# Conclusion générale & Perspectives



# CONCLUSION GENERALE ET PERSPECTIVES

Au cours des dernières décennies, l'urbanisme a connu une concentration croissante démographique dans les zones urbaines, ce qui a accéléré le processus d'urbanisation principalement en termes de bâtiments, de routes et d'infrastructures. Lorsque les villes sont devenues très peuplées, l'homme s'est dirigé vers la création de nouvelles villes ou vers l'expansion des villes vers l'extérieur dans un réseau de périphéries.

La croissance des zones urbaines pousse les pouvoirs publics et d'autres acteurs à coopérer et à planifier en permanence pour une vie meilleure. Néanmoins, la planification de projets urbains demeure une tâche fastidieuse qui demande beaucoup d'efforts, en particulier lorsqu'il s'agit de satisfaire les différents besoins urbains tout en respectant les contraintes (les ressources, la qualité et le temps accordé).

En fait, les meilleurs plans urbains doivent répondre aux objectifs attendus par une bonne affectation des projets urbains aux zones appropriées en gardant la cohérence et l'intégrité du territoire.

Finalement, on conclut par un bilan récapitulatif des travaux réalisés et des perspectives envisageables au terme de cette recherche.

## 1. Bilan de la thèse

Face aux contraintes liées à la spécificité du domaine urbain et en tenant compte de sa dimension et de sa complexité, nous avons utilisé l'approche holonique pour modéliser notre système d'aide à la décision pour la planification des projets urbains, dont les agents holoniques sont supportés par des techniques intelligentes qui intègrent des algorithmes méta-heuristiques et d'en bénéficier dans l'optimisation des plans des projets urbains.

Non seulement le choix de la méthode appropriée suffit pour la résolution d'un problème d'optimisation à grande échelle, mais cela nécessite également une bonne formulation, une modélisation pertinente et une organisation adéquate de ses composants.

Bien qu'un holon puisse être à la fois une entité et une organisation, ceci nous a permis de recourir aux systèmes multi-agents holoniques (SMAH) afin de faire cohabiter les différentes composantes et niveaux d'abstraction du système urbain.

Etant donné que la planification des projets urbains peut se ramener à un problème du sac-à-dos, nous avons développé un nouvel algorithme d'optimisation des colonies de fourmis appelé BKPACS, pour générer des plans optimaux au niveau de chaque holon de base représentant une zone urbaine. A partir du niveau bas jusqu'au au niveau supérieur de l'holarchie, l'algorithme principal H-MACO coordonne entre les holons de différents niveaux, de manière à produire des plans optimaux globaux.

Les travaux réalisés dans cette thèse ont porté sur la proposition d'un système intelligent d'aide à la décision pour la planification des projets urbains. Ce système destiné à assister la communauté urbaine dans le choix des projets de développement qui correspondent le

mieux aux zones urbaines sans altérer l'intégrité global du territoire. Notre contribution se résume sur les points suivants :

Commençant par une introduction générale, dans laquelle nous avons présenté le contexte de travail, les motivations de la recherche, la problématique, une esquisse générale de la méthodologie utilisée, et les objectifs visés.

Le premier chapitre avait pour objectif d'introduire les concepts de base de la planification urbaine, son utilité, ses composantes, ainsi que la démarche du projet urbain sur lesquels nous avons fixé notre cadre d'étude. Ce chapitre se conclut par la mise en évidence d'une problématique globale montrant l'importance d'introduire des techniques intelligentes dans le processus de la planification des projets urbains.

Le deuxième chapitre a exposé les différentes techniques et systèmes intelligents d'aide à la décision (SIAD) y compris les méthodes d'optimisation telles que les approches heuristiques et méta-heuristiques. Nous avons également mis l'accent sur les approches d'optimisation dans les SIAD et leur utilité dans la résolution des problèmes décisionnels, combinatoires et complexes, puis nous avons abordé les problèmes classiques d'optimisation combinatoire, notamment le problème de sac-à-dos et ses variantes qui nous a servi à modéliser le problème de la planification des projets urbains. Ensuite, nous avons présenté en détail l'optimisation par les colonies de fourmis comme étant une technique prometteuse dans les systèmes intelligents d'aide à la décision. Ce chapitre a été enrichi par un état de l'art couvrant les travaux connexes.

Le troisième chapitre est consacré aux contributions en mettant en valeur l'approche holonique, sur laquelle est fondée le système multi-agents holonique (SMAH), ainsi que les techniques d'optimisation méta-heuristiques par colonies de fourmis, afin de mettre en place un système d'aide à la décision pour la planification des projets urbains. Ce chapitre comprend davantage un support technique d'algorithmes (BKPCS, H-MACO) développés spécialement en faveur des agents holoniques.

Dans le quatrième chapitre, nous avons effectué une analyse et une évaluation du système en question. Pour tester les algorithmes BKPCS et H-MACO, et mettre en évidence l'utilisabilité de l'approche holonique conjointement avec les techniques d'optimisation méta-heuristiques par colonies de fourmis dans la planification des projets urbains, nous avons développé un prototype permettant d'évaluer les fonctionnalités de notre système.

Les expérimentations menées sur les cas d'étude ont montré l'efficacité et la performance des algorithmes proposés pour fournir des plans urbains optimaux dans un temps de calcul réduit en les comparant à d'autres algorithmes heuristiques et méta-heuristiques courants notamment les algorithmes génétiques.

## **2. Perspectives de la recherche**

Arrivant à ce stade, notre étude nous a permis de réaliser les objectifs tracés au préalable, qui tournent autour de l'optimisation des plans de projets urbains, à travers l'élaboration d'un système intelligent d'aide à la décision conçu particulièrement pour la planification

urbaine. La modélisation de notre système par l'approche holonique a montré que les agents holoniques possédant la particularité d'être à la fois autonomes et dépendants, et par leur organisation, fournissent des solutions robustes et efficaces permettant de traiter des problèmes complexes et à grande échelle.

Les résultats satisfaisants obtenus dans nos expérimentations, nous incitent à poursuivre le développement des travaux présentés dans cette thèse. Ainsi les perspectives envisagées sont les suivantes :

- Dans cette thèse, nous avons proposé un prototype pour tester les algorithmes d'optimisation développés, via une simulation sur les cas d'étude, néanmoins cela n'est pas suffisant pour attester la validité de notre système et de son architecture. Ce qui ouvre des perspectives de pouvoir les appliquer sur d'autres problèmes afin de tester leur généralité et l'applicabilité.
- Améliorer le modèle organisationnel de notre système d'aide à la décision, en donnant plus de privilèges aux agents holoniques pour qu'ils puissent s'auto-organiser en fonction du secteur d'activité et de la nature du territoire urbain, citons à titre d'exemple les holons chargés de planifier des projets de transport n'ont pas forcément les mêmes prérogatives que les holons chargés de planifier les projets de santé.
- Etendre la formulation du problème de la planification des projets urbains au niveau local, en modifiant la fonction objectif qui utilise uniquement des préférences intrinsèques des projets, de telle façon à introduire la relation préférentielle où un gain supplémentaire lorsque deux projets sont choisis simultanément. Par exemple, lorsqu'on souhaite maximiser la qualité d'un projet de logements, on peut comprendre qu'il est préférable d'achever en parallèle son éclairage public plutôt que d'instaurer un stade de proximité.
- Généraliser le modèle proposé de la planification des projets urbains pour l'utiliser dans d'autres domaines d'application notamment dans la planification des projets agricoles, la régulation du marché de l'emploi, le transport, gestion du ramassage des déchets ménagers, et la gestion des ressources énergétiques, hydriques, et alimentaires etc.
- Développer davantage notre modèle de négociation lors de l'élaboration des plans urbains par les agents holoniques en cas de désaccord entre les parties prenantes, en s'inspirant des modèles d'interaction des SMA.
- Il serait bien d'introduire d'autres fonctionnalités dans notre SIAD, permettant la programmation des projets urbains c'est-à-dire l'ordre dans lequel les projets seront réalisés, ainsi que la programmation par étapes (*multi-étages programming*), qui supporte la planification consécutive de plusieurs programmes de développement.
- Finalement, au niveau des holons de planification, il sera intéressant d'intégrer d'autres techniques intelligentes comme les réseaux de neurones artificiels, les ensembles d'approximation, et l'apprentissage par renforcement afin de prédire des plans de projets urbains en se basant sur les plans urbains précédemment retenus. Et par conséquent enrichir davantage l'architecture de notre système.



## Bibliographie

- [Abdmouleh et al. (2017)] Zeineb Abdmouleh, Adel Gastli, Lazhar Ben-Brahim, Mohamed Haouari, and Nasser Ahmed Al-Emadi. Review of optimization techniques applied for the integration of distributed generation from renewable energy sources. *Renewable Energy*, 113: 266–280, 2017.
- [Adam et al. (1999)] E Adam, R Mandiau, and C Kolski. Approche holonique de modelisation d'une organisation orientee workflow : Sohtco. In M p Gleizes and P Marcenac, editors, *Ingenierie des Systemes multi-agents*, pages 121–134, Paris, janvier 1999. HERMES Science publications.
- [Adam and Kolski (1999)] Emmanuel Adam and Christophe Kolski. Etude comparative de méthodes du génie logiciel en vue du développement de systèmes interactifs dans les processus administratifs complexes. *Génie Logiciel*, 49: 40–54, 1999.
- [Albert et al. (2003)] Martin Albert, Thomas Längle, Heinz Wörn, Michele Capobianco, and Attilio Brighenti. Multi-agent systems for industrial diagnostics. *IFAC Proceedings Volumes*, 36 (5): 459–464, 2003.
- [Alexander (1992)] Ernest R Alexander. *Approaches to planning: Introducing current planning theories, concepts, and issues*. Taylor & Francis, 1992.
- [Ammi (2019)] Houssameddine Ammi. *Villes et développement économique en Algérie*. PhD thesis, Toulon, 2019.
- [Andonov et al. (2000)] Rumen Andonov, Vincent Poirriez, and Sanjay Rajopadhye. Unbounded knapsack problem: Dynamic programming revisited. *European Journal of Operational Research*, 123 (2): 394–407, 2000.
- [Arai (1997)] Tet al Arai. Holonic storage: an assembly and storage cell by manipulation using environment. In *Proc. 29th CIRP Int. Seminar on Manufacturing Systems, Osaka*, volume 221, 1997.
- [Ascher (1991)] François Ascher. Projet public et réalisations privées. Le renouveau de la planification des villes. In *Les annales de la recherche urbaine*, volume 51, pages 5–15. Centre de Recherche d'Urbanisme, 1991. doi: 10.3406/aru.1991.1594.
- [Ascher (1995)] François Ascher. *Métapolis: ou l'avenir des villes*. Odile Jacob, 1995.
- [Athanasiadis and Mitkas (2004)] Ioannis N Athanasiadis and Pericles A Mitkas. An agent-based intelligent environmental monitoring system. *Management of Environmental Quality: An International Journal*, 2004.
- [Ausiello et al. (1995)] Giorgio Ausiello, Pierluigi Crescenzi, and Marco Protasi. Approximate solution of np optimization problems. *Theoretical Computer Science*, 150 (1): 1–55, 1995.
- [Avitabile (2005)] Alain Avitabile. *La mise en scène du projet urbain. Pour une structuration des démarches*. 2005.
- [Babiceanu and Chen (2006)] Radu F Babiceanu and F Frank Chen. Development and applications of holonic manufacturing systems: a survey. *Journal of Intelligent Manufacturing*, 17 (1): 111–131, 2006.
- [Bach and Perea (2020)] Christian W Bach and Andrés Perea. Generalized nash equilibrium without common belief in rationality. *Economics Letters*, 186: 108526, 2020. doi: <https://doi.org/10.1016/j.econlet.2019.108526>.
- [Bacqué and Gauthier (2011)] Marie-Hélène Bacqué and Mario Gauthier. Participation, urbanisme et études urbaines. *Participations*, (1): 36–66, 2011.

- [Bakos and Dumitrascu (2017)] Levente Bakos and Dănut Dumitrascu. Holonic crisis handling model for corporate sustainability. *Sustainability*, 9 (12): 2266, 2017.
- [Beatley et al. (1997)] Timothy Beatley, Beatley Timothy, and Kristy Manning. *The ecology of place: Planning for environment, economy, and community*. Island Press, 1997.
- [Belhédi (1989)] Amor Belhédi. Le découpage administratif en tunisie. *Revue de géographie du Maroc*, 13 (2): 3–25, 1989.
- [Benaïssa (2013)] Ezzeddine Benaïssa. *Plate-forme intelligente pour la chane logistique: approche basée sur un système multi-agents et les services web sémantiques, cas du transport multimodal des marchandises*. PhD thesis, Le Havre, 2013.
- [Bendakir (2016)] Soumia Bendakir. *Une approche basé agent pour le développement du processus d'ingénierie des exigences*. PhD thesis, Université Constantine 2 - Abdelhamid Mehri, 2016.
- [Benyamina (2013)] Ahmed Benyamina. *Application des algorithmes de colonies de fourmis pour l'optimisation et la classification des images*. PhD thesis, USTO (MB), 2013.
- [Berezowska-Azzag et al. (2012)] Ewa Berezowska-Azzag, Isma Abdelatif, Nadia Akrou, and Rabah Halimi. Approche du projet urbain par la mise en attractivité d'un territoire. *HS 04*: 80–88, 12 2012.
- [Bieszczad et al. (1998)] Andrzej Bieszczad, Bernard Pagurek, and Tony White. Mobile agents for network management. *IEEE Communications Surveys*, 1 (1): 2–9, 1998.
- [Binmore (1990)] K Binmore. *Essays on the foundations of game theory*, oxford: Basil blackwell, 1990.
- [Bohanec (2003)] Marko Bohanec. Decision support. In *Data mining and decision support*, pages 23–35. Springer, 2003.
- [Bonabeau et al. (1999)] Eric Bonabeau, Marco Dorigo, Directeur de Recherches Du Fnrs Marco, Guy Theraulaz, Guy Théraulaz, et al. *Swarm intelligence: from natural to artificial systems*. Number 1. Oxford university press, 1999.
- [Bouchareb (2011)] Abdelouahab Bouchareb. Projet urbain. *Cours: Master 1 académique, Université Mentouri-Constantine*, 2011.
- [Breuillé et al. (2019)] Marie-Laure Breuillé, Camille Grivault, Julie Le Gallo, and Renaud Le Goix. Impact de la densification sur les coûts des infrastructures et services publics. *Revue économique*, 70 (3): 345–373, 2019.
- [Bürckert et al. (2000)] Hans-Jürgen Bürckert, Klaus Fischer, and Gero Vierke. Holonic transport scheduling with teletruck. *Applied Artificial Intelligence*, 14 (7): 697–725, 2000.
- [Cai et al. (2019)] Xingjuan Cai, Jiangjiang Zhang, Hao Liang, Lei Wang, and Qidi Wu. An ensemble bat algorithm for large-scale optimization. *International Journal of Machine Learning and Cybernetics*, 10 (11): 3099–3113, 2019.
- [Calabrese et al. (2010)] Marco Calabrese, Alberto Amato, Vincenzo Di Lecce, and Vincenzo Piuri. Hierarchical-granularity holonic modelling. *Journal of Ambient Intelligence and Humanized Computing*, 1 (3): 199–209, 2010.
- [Canning (1992)] David Canning. Rationality, computability, and nash equilibrium. *Econometrica: Journal of the Econometric Society*, pages 877–888, 1992. doi: <https://doi.org/10.2307/2951570>.
- [Capel (1975)] Horacio Capel. L'image de la ville et le comportement spatial des citadins. *L'Espace géographique*, pages 73–80, 1975.
- [Chebli (2011)] Nora Chebli. Ressources et développement local de la commune d'ain melouk. *Mémoire Pour l'obtention du diplôme de magister*, 2011.

- [Chen and Pham (2000)] Guanrong Chen and Trung Tat Pham. *Introduction to fuzzy sets, fuzzy logic, and fuzzy control systems*. CRC press, 2000.
- [Chen and Ichalkaranje (2013)] Zhengxin Chen and Nikhil Ichalkaranje. *Intelligent agents and their applications*, volume 98. Physica, 2013.
- [Clément and Guth (1995)] Pierre Clément and Sabine Guth. De la densité qui tue à la densité qui paye. la densité urbaine comme règle et médiateur entre politique et projet. In *Les Annales de la recherche urbaine*, volume 67, pages 72–83. Centre de Recherche d'Urbanisme, 1995.
- [Colorni et al. (1992)] Alberto Colorni, Marco Dorigo, Vittorio Maniezzo, et al. An investigation of some properties of an "ant algorithm". In *Ppsn*, volume 92, 1992.
- [Cortés et al. (2000)] Ulises Cortés, Miquel Sànchez-Marrè, Luigi Ceccaroni, Ignasi R-Roda, and Manel Poch. Artificial intelligence and environmental decision support systems. *Applied intelligence*, 13 (1): 77–91, 2000.
- [Courcier (2005)] Sabine Courcier. Vers une définition du projet urbain, la planification du réaménagement du Vieux-Port de Montréal. *Canadian Journal of Urban Research*, pages 57–80, 2005.
- [Courdier et al. (1998)] Rémy Courdier, Pierre Marcenac, and Sylvain Giroux. Un processus de développement en spirale pour la simulation multi-agents. *L'Objet*, 4 (1): 73–86, 1998.
- [Dantzig (1957)] George B Dantzig. Discrete-variable extremum problems. *Operations research*, 5 (2): 266–288, 1957.
- [Das (2016)] TK Das. Intelligent techniques in decision making: a survey. *Indian Journal of Science and Technology*, 9 (12): 1–6, 2016.
- [De Leoni et al. (2016)] Massimiliano De Leoni, Wil MP van der Aalst, and Marcus Dees. A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. *Information Systems*, 56: 235–257, 2016.
- [Decker et al. (1997)] Keith Decker, Katia Sycara, and Mike Williamson. Middle-agents for the internet. In *IJCAI (1)*, pages 578–583, 1997.
- [Deneubourg (1990)] Jean-Louis Deneubourg. Swarm-made architectures. *Toward a practice of autonomous systems*, pages 123–133, 1990.
- [Deneubourg et al. (1991)] Jean-Louis Deneubourg, Simon Goss, Nigel Franks, Ana Sendova-Franks, Claire Detrain, and Laeticia Chrétien. The dynamics of collective sorting robot-like ants and ant-like robots. In *Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats*, pages 356–363, 1991.
- [Dind (2011)] Jean-Philippe Dind. La gestion de projets urbains. Projets d'aménagement concentrés dans des secteurs déjà bâtis : exemples en Suisse Romande. *Disponible sur <http://www.unil.ch/ouvdd>*, 2011.
- [Dorigo and Gambardella (1996)] Marco Dorigo and Luca Maria Gambardella. A study of some properties of ant-q. In *International Conference on Parallel Problem Solving from Nature*, pages 656–665. Springer, 1996.
- [Dorigo and Gambardella (1997)] Marco Dorigo and Luca Maria Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, 1 (1): 53–66, 1997. doi : <https://doi.org/10.1109/4235.585892>.
- [Dorigo et al. (1991)] Marco Dorigo, Alberto Colorni, and Vittorio Maniezzo. Distributed optimization by ant colonies, 1991.

- [Dorigo et al. (1996)] Marco Dorigo, Vittorio Maniezzo, and Alberto Coloni. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26 (1): 29–41, 1996.
- [Dorigo et al. (2000)] Marco Dorigo, Eric Bonabeau, and Guy Theraulaz. Ant algorithms and stigmergy. *Future Generation Computer Systems*, 16 (8): 851–871, 2000.
- [Dorri et al. (2018)] Ali Dorri, Salil S Kanhere, and Raja Jurdak. Multi-agent systems: A survey. *Ieee Access*, 6: 28573–28593, 2018.
- [Dubois-Maury and Paquot (2010)] J Dubois-Maury and T Paquot. L’abc de l’urbanisme, institut d’urbanisme de paris, 2010.
- [Duffey and Stratford (1989)] Christopher K Duffey and Ray P Stratford. Update of harmonic standard ieee-519: Ieee recommended practices and requirements for harmonic control in electric power systems. *IEEE Transactions on Industry Applications*, 25 (6): 1025–1034, 1989.
- [Dureau (1993)] Françoise Dureau. Pour une approche non fonctionnaliste. *Croissance démographique et urbanisation: politique de peuplement et aménagement du territoire: séminaire international de Rabat (15-17 mai 1990)*, (5): 105, 1993.
- [Emelianoff (2007)] Cyria Emelianoff. La ville durable: l’hypothèse d’un tournant urbanistique en europe. *L’Information géographique*, 71 (3): 48–65, 2007.
- [Epstein (2013)] Renaud Epstein. Politiques territoriales : ce que les appels à projets font aux démarches de projet. *The Tocqueville Review/La revue Tocqueville*, 34 (2): 91–102, 2013.
- [Etzioni and Weld (1994)] Oren Etzioni and Daniel Weld. A softbot-based interface to the internet. *Communications of the ACM*, 37 (7): 72–76, 1994.
- [Fayyad et al. (1996)] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39 (11): 27–34, 1996.
- [Ferber and Weiss (1999)] Jacques Ferber and Gerhard Weiss. *Multi-agent systems: an introduction to distributed artificial intelligence*, volume 1. Addison-Wesley Reading, 1999.
- [Fernandes and Christine (2001)] Corrêa E Silva Fernandes and Kelly Christine. *Systemes multi-agents hybrides: une approche pour la conception de systemes complexes*. PhD thesis, Grenoble 1, 2001.
- [Festa (2014)] Paola Festa. A brief introduction to exact, approximation, and heuristic algorithms for solving hard combinatorial optimization problems. In *2014 16th International Conference on Transparent Optical Networks (ICTON)*, pages 1–20. IEEE, 2014.
- [Fouilhoux (2015)] Pierre Fouilhoux. Optimisation combinatoire: Programmation linéaire et algorithmes. *Université Pierre et Marie Curie*, 2015.
- [Furini et al. (2018)] Fabio Furini, Michele Monaci, and Emiliano Traversi. Exact approaches for the knapsack problem with setups. *Computers & Operations Research*, 90: 208–220, 2018.
- [Gallez and Maksim (2007)] Caroline Gallez and Hanja-Niriana Maksim. À quoi sert la planification urbaine? *Flux*, (3): 49–62, 2007.
- [Gandreau et al. (2018)] David Gandreau, Grégoire Paccoud, Bakonirina Rakotomamonjy, Emmanuelle Robert, Dora Studer, and Joffroy Thierry. Réhabilitation et valorisation du bâti en pisé, 2018.

- [Gaud (2007)] Nicolas A Gaud. *Systèmes multi-agents holoniques: de l'analyse à l'implantation: méta-modèle, méthodologie, et simulation multi-niveaux*. PhD thesis, Besançon, 2007.
- [Gaudin (1985)] Jean-Pierre Gaudin. *L'avenir en plan: technique et politique dans la prévision urbaine, 1900-1930*. Editions Champ Vallon, 1985.
- [Gerber et al. (1999a)] Christian Gerber, Jörg Siekmann, and Gero Vierke. Flexible autonomy in holonic agent systems. In *Proceedings of the 1999 AAAI Spring Symposium on Agents with Adjustable Autonomy*, 1999.
- [Gerber et al. (1999b)] Christian Gerber, Jörg Siekmann, and Gero Vierke. Holonic multi-agent systems. 1999.
- [Ghahramani (2003)] Zoubin Ghahramani. Unsupervised learning. In *Summer School on Machine Learning*, pages 72–112. Springer, 2003.
- [Giret and Botti (2004)] Adriana Giret and Vicente Botti. Holons and agents. *Journal of intelligent manufacturing*, 15 (5): 645–659, 2004.
- [Giudici et al. (2017)] Alberto Giudici, Pascal Halffmann, Stefan Ruzika, and Clemens Thielen. Approximation schemes for the parametric knapsack problem. *Information Processing Letters*, 120: 11–15, 2017.
- [Glorfeld and Hardgrave (1996)] L. W. Glorfeld and B. C. Hardgrave. An improved method for developing neural networks: The case of evaluating commercial loan creditworthiness. *Computer Operation Research*, 23 (10): 933–944, 1996.
- [Glover (1986)] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers operations research*, 13 (5): 533–549, 1986.
- [Glover and Laguna (1998)] Fred Glover and Manuel Laguna. Tabu search. In *Handbook of combinatorial optimization*, pages 2093–2229. Springer, 1998.
- [Goodfellow et al. (2016)] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.
- [Gordon et al. (2009)] Ascelin Gordon, David Simondson, Matt White, Atte Moilanen, and Sarah Adine Bekessy. Integrating conservation planning and landuse planning in urban landscapes. *Landscape and urban planning*, 91 (4): 183–194, 2009.
- [Gray et al. (2019)] Justin S Gray, John T Hwang, Joaquim RRA Martins, Kenneth T Moore, and Bret A Naylor. Openmdao: An open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*, 59 (4): 1075–1104, 2019. doi : <https://doi.org/10.1007/s00158-019-02211-z>.
- [Greco et al. (2001)] Salvatore Greco, Benedetto Matarazzo, and Roman Slowinski. Rough sets theory for multicriteria decision analysis. *European journal of operational research*, 129 (1): 1–47, 2001.
- [Gwiazdzinski (2015)] Luc Gwiazdzinski. *Le design territorial nouvelle frontière de l'acte public*, 2015.
- [Haddar et al. (2016)] Boukthir Haddar, Mahdi Khemakhem, Sad Hanafi, and Christophe Wilbaut. A hybrid quantum particle swarm optimization for the multidimensional knapsack problem. *Engineering Applications of Artificial Intelligence*, 55: 1–13, 2016.
- [He et al. (2019)] Cheng He, Lianghao Li, Ye Tian, Xingyi Zhang, Ran Cheng, Yaochu Jin, and Xin Yao. Accelerating large-scale multiobjective optimization via problem reformulation. *IEEE Transactions on Evolutionary Computation*, 23 (6): 949–961, 2019.
- [Hojny et al. (2019)] Christopher Hojny, Tristan Gally, Oliver Habeck, Hendrik Lüthen, Frederic Matter, Marc E Pfetsch, and Andreas Schmitt. Knapsack polytopes: a survey. *Annals of Operations Research*, pages 1–49, 2019.

- [Holland (1995)] John Henry Holland. *Hidden order how adaptation builds complexity*. Number 003.7 H6. 1995.
- [Huang et al. (2014)] Gao Huang, Shiji Song, Jatinder ND Gupta, and Cheng Wu. Semi-supervised and unsupervised extreme learning machines. *IEEE transactions on cybernetics*, 44 (12): 2405–2417, 2014.
- [Huhns and Singh (1998)] Michael N Huhns and Munindar P Singh. *Readings in agents*. Morgan Kaufmann, 1998.
- [Indriago et al. (2016)] Carlos Indriago, Olivier Cardin, Naly Rakoto, Pierre Castagna, and Edgar Chacòn. H2cm: A holonic architecture for flexible hybrid control systems. *Computers in industry*, 77: 15–28, 2016.
- [Ingallina (2001)] Patrizia Ingallina. *Le projet urbain*. Presses universitaires de France, 2001.
- [Jennings (1995)] Nicholas R Jennings. Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial intelligence*, 75 (2): 195–240, 1995.
- [Jennings et al. (1998)] Nick Jennings, Nicholas R Jennings, and Michael J Wooldridge. *Agent technology: foundations, applications, and markets*. Springer Science & Business Media, 1998.
- [Jenzer (1996)] Mélanie Jenzer. Les systèmes d'information géographique. 1996.
- [Kaiser et al. (1995)] Edward J Kaiser, David R Godschalk, and F Stuart Chapin. *Urban land use planning*, volume 4. University of Illinois press Urbana, 1995.
- [Kantamneni et al. (2015)] Abhilash Kantamneni, Laura E Brown, Gordon Parker, and Wayne W Weaver. Survey of multi-agent systems for microgrid control. *Engineering applications of artificial intelligence*, 45: 192–203, 2015.
- [Karp (1972)] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.
- [Keeney et al. (1993)] Ralph L Keeney, Howard Raiffa, et al. *Decisions with multiple objectives: preferences and value trade-offs*. Cambridge university press, 1993.
- [Kellerer et al. (2004)] Hans Kellerer, Ulrich Pferschy, and David Pisinger. Multidimensional knapsack problems. In *Knapsack problems*, pages 235–283. Springer, 2004.
- [Khelifa and Laouar (2018)] Boudjemaa Khelifa and Mohamed Ridda Laouar. Urban projects planning by multi-objective ant colony optimization algorithm. In *Proceedings of the 8th International Conference on Information Systems and Technologies*, pages 1–6, 2018.
- [Khelifa and Laouar (2020)] Boudjemaa Khelifa and Mohamed Ridda Laouar. A holonic intelligent decision support system for urban project planning by ant colony optimization algorithm. *Applied Soft Computing*, 96: 106621, 2020.
- [Khelifa et al. (2018)] Boudjemaa Khelifa, Mohamed Ridda Laouar, and Sean Eom. Towards an intelligent integrated system for urban planning using gis and cloud computing. In *International Conference on Decision Support System Technology*, pages 26–37. Springer, 2018.
- [Khoong (1995)] CM Khoong. Decision support systems: an extended research agenda. *Omega*, 23 (2): 221–229, 1995.
- [Kirkpatrick et al. (1983)] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220 (4598): 671–680, 1983.
- [Kleindorfer et al. (1993)] Paul R Kleindorfer, Howard Kunreuther, Howard G Kunreuther, and Paul JH Schoemaker. *Decision sciences: An integrative perspective*. Cambridge University Press, 1993.

- [Klusch (2001)] Matthias Klusch. Information agents: Theory and applications guest editor's introduction. *Int. J. Cooperative Inf. Syst.*, 10 (1-2): 51–56, 2001.
- [Koestler (1967)] Arthur Koestler. The ghost in the machine London. *Pan*, page 73, 1967.
- [Koestler (1969)] Arthur Koestler. Beyond atomism and holism - the concept of the holon. *The rules of the game: Cross-disciplinary essays on models in scholarly thought*, pages 233–247, 1969.
- [Kotsiantis et al. (2007)] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160 (1): 3–24, 2007.
- [Kramer (2017)] Oliver Kramer. *Genetic algorithm essentials*, volume 679. Springer, 2017.
- [Krause et al. (2013)] Jonas Krause, Jelson Cordeiro, Rafael Stubs Parpinelli, and Heitor Silverio Lopes. A survey of swarm algorithms applied to discrete optimization problems. In *Swarm Intelligence and Bio-Inspired Computation*, pages 169–191. Elsevier, 2013.
- [Kwon et al. (2005)] Ohbyung Kwon, Keedong Yoo, and Euiho Suh. UbiDSS: a proactive intelligent decision support system as an expert system deploying ubiquitous computing technologies. *Expert systems with applications*, 28 (1): 149–161, 2005.
- [Laabadi et al. (2018)] Soukaina Laabadi, Mohamed Naimi, Hassan El Amri, Boujemâa Achchab, et al. The 0/1 multidimensional knapsack problem and its variants: a survey of practical models and heuristic approaches. *American Journal of Operations Research*, 8 (05): 395, 2018.
- [Lakhdar Hamina and Abbas (2015)] Youcef Lakhdar Hamina and Leila Abbas. Evolution des instruments de planification spatiale et de gestion urbaine en Algérie. *Cinq Continents*, 5 (11): 104–129, 2015.
- [Laurini (2003)] Robert Laurini. Information systems for urban planning: A hypermedia collaborative approach. *PLANNING THEORY AND PRACTICE*, 4 (2): 236–236, 2003.
- [Lecoq (2017)] Matthias Lecoq. *Production de la ville: innovation des processus participatifs et citoyenneté urbaine*. 2017.
- [LeCun et al. (2015)] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521 (7553): 436–444, 2015.
- [Legrand (1992)] Michel Legrand. Diversité de l'habitat, mixité urbaine et planification urbaine. *Mars*, page 110p, 1992.
- [Legrand and Meunier-Chabert (2003)] Michel Legrand and Martine Meunier-Chabert. Diversité de l'habitat, mixité urbaine et planification urbaine: enquête sur la mise en oeuvre des principes de la loi d'orientation pour la ville (lov). 2003.
- [Leitao (2009)] Paulo Leitao. Holonic rationale and bio-inspiration on design of complex emergent and evolvable systems. In *Transactions on Large-Scale Data-and Knowledge-Centered Systems I*, pages 243–266. Springer, 2009.
- [Li et al. (2019)] Jiang Li, Lihong Guo, Yan Li, and Chang Liu. Enhancing elephant herding optimization with novel individual updating strategies for large-scale optimization problems. *Mathematics*, 7 (5): 395, 2019.
- [Liu and Zhu (2008)] Guilong Liu and William Zhu. The algebraic structures of generalized rough set theory. *Information Sciences*, 178 (21): 4105–4113, 2008.

- [Malviya (2017)] Tarun Malviya. A multi-dimensional business strategy model for effective decision making: A holonic representation of enterprises. In *2017 IEEE International Systems Engineering Symposium (ISSE)*, pages 1–6. IEEE, 2017.
- [Mangin and Panerai (1999)] David Mangin and Philippe Panerai. *Projet urbain*. Parenthèses Marseille, 1999.
- [Maniezzo and Carbonaro (2002)] Vittorio Maniezzo and Antonella Carbonaro. Ant colony optimization: an overview. In *Essays and surveys in metaheuristics*, pages 469–492. Springer, 2002.
- [Mark et al. (2002)] Vladimr Mark, Michal Pechoucek, and Olga Štěpánková. Organization of social knowledge in multi-agent systems. *Integrated Computer-Aided Engineering*, 9 (3): 195–206, 2002.
- [Mathieu and Guermond (2005)] Nicole Mathieu and Yves Guermond. *La Ville durable, du politique au scientifique*. Cemagref, 2005.
- [Maturana et al. (1999)] Francisco Maturana, Weiming Shen, and Douglas H Norrie. Metamorph: an adaptive agent-based architecture for intelligent manufacturing. *International Journal of Production Research*, 37 (10): 2159–2173, 1999.
- [McFarlane and Bussmann (2000)] Duncan C McFarlane and Stefan Bussmann. Developments in holonic production planning and control. *Production Planning & Control*, 11 (6): 522–536, 2000.
- [McLoughlin (1969)] J Brian McLoughlin. *Urban & regional planning: a systems approach*. Faber and Faber, 1969.
- [Mella (2009)] Piero Mella. The holonic revolution. *Holons, Holarchies and Holonic*, 2009.
- [Merlin (1996)] Pierre Merlin. *L’urbanisme: L’Urbanisme*. Presses universitaires de France, 1996.
- [Mesarovic et al. (2000)] Mihajlo D Mesarovic, Donald Macko, and Yasuhiko Takahara. *Theory of hierarchical, multilevel, systems*. Elsevier, 2000.
- [Misevicius (2003)] Alfonsas Misevicius. A modified simulated annealing algorithm for the quadratic assignment problem. *Informatica*, 14 (4): 497–514, 2003.
- [Montedoro and Costa (2004)] LAURA Montedoro and A Costa. La ville rationaliste. urbanisme et architecture à modène 1931-1965. 2004.
- [Nash (1951)] John Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951. doi : <https://doi.org/10.2307/1969529>.
- [Offner (2002)] Jean-Marc Offner. X. les transports urbains: entre secteurs, réseaux et territoires. *Annuaire des collectivités locales*, 22 (1): 169–183, 2002.
- [Okamoto et al. (2008)] Steven Okamoto, Paul Scerri, and Katia P Sycara. The impact of vertical specialization on hierarchical multi-agent systems. In *AAAI*, pages 138–143, 2008.
- [Padgham and Winikoff (2005)] Lin Padgham and Michael Winikoff. *Developing intelligent agent systems: A practical guide*, volume 13. John Wiley & Sons, 2005.
- [Paetzold and Rollet (2011)] Heinz Paetzold and Brigitte Rollet. Esthétique du design urbain. *Diogene*, (1): 91–103, 2011.
- [Paquot and Younès (2010)] Thierry Paquot and Chris Younès. Philosophie de l’environnement et milieux urbains. *Lectures, Les livres*, 2010.
- [Paris et al. (2008)] Benjamin Paris, Julien Eynard, Grégory François, Thierry Talbert, Adama Traore, and Frédérik Thiéry. Gestion des ressources énergétiques d’un bâtiment: contrôle flou. 2008.



- [Pawlak (1982)] Zdzisaw Pawlak. Rough sets. *International journal of computer & information sciences*, 11 (5): 341–356, 1982.
- [Pawlak and Skowron (2007)] Zdzisaw Pawlak and Andrzej Skowron. Rough sets: some extensions. *Information sciences*, 177 (1): 28–40, 2007.
- [Peng et al. (1996)] Tian Peng, Wang Huanchen, and Zhang Dongme. Simulated annealing for the quadratic assignment problem: A further study. *Computers & industrial engineering*, 31 (3-4): 925–928, 1996.
- [Peng et al. (1998)] Yun Peng, Tim Finin, Yannis K Labrou, Bill Chu, J Long, William Tolone, Akram Boughannam, et al. A multi-agent system for enterprise integration. *International Journal of Agile Manufacturing*, 1 (2): 213–229, 1998.
- [Pétrowski and Ben-Hamida (2017)] Alain Pétrowski and Sana Ben-Hamida. *Evolutionary Algorithms*. John Wiley & Sons, 2017.
- [Phan and Varenne (2010)] Denis Phan and Franck Varenne. Agent-based models and simulations in economics and social sciences: from conceptual exploration to distinct ways of experimenting. *Journal of Artificial Societies and Social Simulation*, 13 (1): 5, 2010.
- [Phillips-Wren (2008)] Gloria E Phillips-Wren. Intelligent agents in decision support systems. In *Encyclopedia of Decision Making and Decision Support Technologies*, pages 505–513. IGI Global, 2008.
- [Pinson (2005)] Gilles Pinson. L'idéologie des projets urbains. 2005.
- [Pinson (2006)] Gilles Pinson. Opaques métropoles. Dans Bernard Jouve (dir.), *Cours politiques urbaines, Vaulin, Université de Lyon-ENTPE*, 374: 59–99, 2006. ISSN 0033-0884. doi: 10.3917/pro.374.0047.
- [Pinson (2009)] Gilles Pinson. Gouverner par projet. *Urbanisme et gouvernance des villes européennes, Paris, Presses de Sciences Po*, 2009.
- [Pornon (2015)] Henri Pornon. *SIG: la dimension géographique du système d'information*. Dunod, 2015.
- [Power (2007)] Daniel J Power. A brief history of decision support systems. *DSSResources.com*, 3, 2007.
- [Rabinovich and Navez-Bouchanine (2005)] Adriana Rabinovich and Françoise Navez-Bouchanine. Projet urbain: entre innovation et tradition dans l'action urbaine. In *intervention lors du colloque Développement urbain durable, gestion des ressources et gouvernance à Lausanne, co-organisé par l'école polytechnique de Lausanne et institut de*, 2005.
- [Rafieian et al. (2009)] Mojtaba Rafieian, Ali Asgari, and Z Asgarizadeh. Citizen satisfaction evaluation of urban residential environment. 2009.
- [Raza and Qamar (2017)] Muhammad Summair Raza and Usman Qamar. Rough set-based feature selection techniques. In *Understanding and Using Rough Set Based Feature Selection: Concepts, Techniques and Applications*, pages 109–129. Springer, 2017.
- [Rissino and Lambert-Torres (2009)] Silvia Rissino and Germano Lambert-Torres. Rough set theory - fundamental concepts, principals, data extraction, and applications. In *Data mining and knowledge discovery in real life applications*. IntechOpen, 2009.
- [Rode (2017)] Sylvain Rode. Reconquérir les cours d'eau pour aménager la ville. *Cybergeog: European Journal of Geography*, 2017.
- [Rodriguez (2006)] Sebastian Rodriguez. *From analysis to design of Holonic Multi-Agent Systems: a Framework, methodological guidelines and applications*. PhD thesis, Besançon, 2006.

- [Rumbaugh et al. (1991)] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William E. Lorenzen, et al. *Object-oriented modeling and design*, volume 199. Prentice-hall Englewood Cliffs, NJ, 1991.
- [Russell and Norvig (2016)] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.
- [Saddem and Hmida (2014)] R Saddem and F Ben Hmida. Multiagent architecture for online diagnosis of discrete event systems using causal temporal signature approach. In *2014 15th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pages 429–434. IEEE, 2014.
- [Saidouni (2001)] Maouia Saidouni. Elements d’introduction à l’urbanisme. *Casbah Edition*, 2001.
- [Salazar et al. (2015)] Oscar M Salazar, Demetrio A Ovalle, and Néstor D Duque. Adaptive and personalized educational ubiquitous multi-agent system using context-awareness services and mobile devices. In *International Conference on Learning and Collaboration Technologies*, pages 301–312. Springer, 2015.
- [Salles (2006)] Denis Salles. Les défis de l’<sup>TM</sup>environnement. *Démocratie et efficacité*, 2006.
- [Samuel (1967a)] Arthur L Samuel. 1959. some studies on machine learning using the game of checkers. *IBM Journal of Research and Development*, 3: 211–229, 1967.
- [Samuel (1967b)] Arthur L Samuel. Some studies in machine learning using the game of checkers. II-recent progress. *IBM Journal of Research and Development*, 11 (6): 601–617, 1967.
- [Sathya and Abraham (2013)] R Sathya and Annamma Abraham. Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, 2 (2): 34–38, 2013.
- [Selman and Gomes (2006)] Bart Selman and Carla P Gomes. Hill-climbing search. *Encyclopedia of cognitive science*, 81: 82, 2006.
- [Selvi and Umarani (2010)] V Selvi and R Umarani. Comparative analysis of ant colony and particle swarm optimization techniques. *International Journal of Computer Applications*, 5 (4): 1–6, 2010.
- [Sénécal et al. (2016)] Gilles Sénécal, Geneviève Cloutier, and Christian Bizier. *Revitalisation urbaine et concertation de quartier*. Presses de l’Université Laval, 2016.
- [Shim et al. (2002)] Jung P Shim, Merrill Warkentin, James F Courtney, Daniel J Power, Ramesh Sharda, and Christer Carlsson. Past, present, and future of decision support technology. *Decision support systems*, 33 (2): 111–126, 2002.
- [Silvano and Paolo (1990)] Martello Silvano and Toth Paolo. Knapsack problems: algorithms and computer implementations, 1990.
- [Simon (1977)] Herbert A Simon. The organization of complex systems. In *Models of discovery*, pages 245–261. Springer, 1977.
- [Smith (2000)] Andrew P. Smith. *Worlds within Worlds. The Hierarchy of Life*. 2000.
- [Su et al. (2011)] Kehua Su, Jie Li, and Hongbo Fu. Smart city and the applications. In *2011 international conference on electronics, communications and control (ICECC)*, pages 1028–1031. IEEE, 2011.
- [Sugi et al. (2003)] Masao Sugi, Yusuke Maeda, Yasumichi Aiyama, Tomokazu Harada, and Tamio Arai. A holonic architecture for easy reconfiguration of robotic assembly systems. *IEEE Transactions on Robotics and Automation*, 19 (3): 457–464, 2003.

- [Sutton and Barto (2018)] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [Sycara (1998)] Katia P Sycara. Multiagent systems. *AI magazine*, 19 (2): 79–79, 1998.
- [Taylor (1998)] Nigel Taylor. *Urban planning theory since 1945*. Sage, 1998.
- [Teo et al. (2012)] Joel SE Teo, Eiichi Taniguchi, and Ali Gul Qureshi. Evaluating city logistics measure in e-commerce with multiagent systems. *Procedia-Social and Behavioral Sciences*, 39: 349–359, 2012.
- [Tian et al. (2019)] Ye Tian, Xiutao Zheng, Xingyi Zhang, and Yaochu Jin. Efficient large-scale multiobjective optimization based on a competitive swarm optimizer. *IEEE Transactions on Cybernetics*, 2019.
- [Topin et al. (1999)] Xavier Topin, Vincent Fourcassié, Marie-Pierre Gleizes, Guy Theraulaz, Christine Régis, and Pierre Glize. Theories and experiments on emergent behaviour: From natural to artificial systems and back. In *Proceedings on European Conference on Cognitive Science, Siena*, 1999.
- [Toussaint and Zimmermann (1998)] Jean-Yves Toussaint and Monique Zimmermann. *Projet urbain: ménager les gens, aménager la ville*, volume 38. Editions Mardaga, 1998.
- [Tweedale et al. (2016)] Jeffrey W Tweedale, Gloria Phillips-Wren, and Lakhmi C Jain. Advances in intelligent decision-making technology support. In *Intelligent Decision Technology Support in Practice*, pages 1–15. Springer, 2016.
- [Tzeng and Huang (2013)] Gwo-Hshiung Tzeng and Jih-Jeng Huang. *Fuzzy multiple objective decision making*. CRC Press, 2013.
- [Ulieru and Geras (2002)] Mihaela Ulieru and Adam Geras. Emergent holarchies for e-health applications: a case in glaucoma diagnosis. In *IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02*, volume 4, pages 2957–2961. IEEE, 2002.
- [Van Brussel et al. (1998)] Hendrik Van Brussel, Jo Wyns, Paul Valckenaers, Luc Bongaerts, and Patrick Peeters. Reference architecture for holonic manufacturing systems: Prosa. *Computers in industry*, 37 (3): 255–274, 1998.
- [Van Laarhoven et al. (1992)] Peter JM Van Laarhoven, Emile HL Aarts, and Jan Karel Lenstra. Job shop scheduling by simulated annealing. *Operations research*, 40 (1): 113–125, 1992.
- [Walczak and Massart (1999)] Beata Walczak and DL Massart. Rough sets theory. *Chemometrics and intelligent laboratory systems*, 47 (1): 1–16, 1999.
- [Wang et al. (2020)] Yuan Wang, Ling Wang, Guangcai Chen, Zhaoquan Cai, Yongquan Zhou, and Lining Xing. An improved ant colony optimization algorithm to the periodic vehicle routing problem with time window and service choice. *Swarm and Evolutionary Computation*, page 100675, 2020.
- [Weiss (1999)] Gerhard Weiss. *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT press, 1999.
- [Wilber (1995)] Ken Wilber. An informal overview of transpersonal studies. *The Journal of Transpersonal Psychology*, 27 (2): 107, 1995.
- [Wooldridge (2009)] Michael Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.
- [Wooldridge and Jennings (1995)] Michael J Wooldridge and Nicholas R Jennings. Intelligent agents: Theory and practice. *The knowledge engineering review*, 10 (2): 115–152, 1995.
- [Yaseen and Al-Slamy (2008)] Saad Ghaleb Yaseen and NM Al-Slamy. Ant colony optimization. *IJCSNS*, 8 (6): 351, 2008.

- [Yin (2012)] Jordan Yin. *Urban planning for dummies*, volume 79. John Wiley & Sons, 2012. doi: 10.1080/01944363.2013.811374.
- [Young (1995)] Michael J Young. Human error and the holon cognitive architecture. *IFAC Proceedings Volumes*, 28 (15): 165–170, 1995.
- [Zadeh (1965)] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8 (3): 338–353, 1965.
- [Zarató et al. (2008)] Pascale Zarató, Jean Pierre Belaud, and G Camilleri. *Collaborative decision making: Perspectives and challenges*, volume 176. IOS Press, 2008.
- [Zenker and Rütter (2014)] Sebastian Zenker and Natascha Rütter. Is satisfaction the key? the role of citizen satisfaction, place attachment and place brand attitude on positive citizenship behavior. *Cities*, 38: 11–17, 2014.
- [Zgurovsky and Pavlov (2018)] Michael Z Zgurovsky and Alexander A Pavlov. *Combinatorial optimization problems in planning and decision making: theory and applications*, volume 173. Springer, 2018.
- [Zhang and Norrie (1999a)] Xiaokun Zhang and Douglas H Norrie. Dynamic reconfiguration of holonic lower level control. In *Proceedings of the Second International Conference on Intelligent Processing and Manufacturing of Materials. IPMM'99 (Cat. No. 99EX296)*, volume 2, pages 887–893. IEEE, 1999.
- [Zhang and Norrie (1999b)] Xiaokun Zhang and Douglas H Norrie. Holonic control at the production and controller levels. In *Proceedings of the 2nd international workshop on intelligent manufacturing systems*, pages 215–224. Citeseer, 1999.
- [Zhang et al. (2020)] Yin Zhang, Gai-Ge Wang, Keqin Li, Wei-Chang Yeh, Muwei Jian, and Junyu Dong. Enhancing moea/d with information feedback models for large-scale many-objective optimization. *Information Sciences*, 2020.
- [Zhu (2005)] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.

# Annexes

## Annexes

### A. Liste des publications

1. KHELIFA, Boudjemaa, LAOUAR, Mohamed Ridda, et EOM, Sean. [Towards an Intelligent Integrated System for Urban Planning Using GIS and Cloud Computing](#). In : *International Conference on Decision Support System Technology*. Springer, Cham, 2018. p. 26-37. [https://doi.org/10.1007/978-3-319-90315-6\\_3](https://doi.org/10.1007/978-3-319-90315-6_3).
2. KHELIFA, Boudjemaa et LAOUAR, Mohamed Ridda. [Urban projects planning by multi-objective ant colony optimization algorithm](#). In : *Proceedings of the 8th International Conference on Information Systems and Technologies*. 2018. p. 1-6. <https://doi.org/10.1145/3200842.3200856>.
3. KHELIFA, Boudjemaa et LAOUAR, Mohamed Ridda. [Multi-agent Reinforcement Learning for Urban Projects Planning](#). In : *Proceedings of the 7th International Conference on Software Engineering and New Technologies*. 2018. p. 1-5. <https://doi.org/10.1145/3330089.3330134>.
4. KHELIFA Boudjemaa et LAOUAR Mohamed Ridda. [A holonic intelligent decision support system for urban project planning by ant colony optimization algorithm](#). *Applied Soft Computing*, 2020, vol. 96, p. 106621. doi: <https://doi.org/10.1016/j.asoc.2020.106621>.

## B. Sources

☞ Code source de l'algorithme d'optimisation multi-objectifs par colonie de fourmis (HMACO + BKPACS) écrit en langage pascal (Environnement : *Embarcadero RAD Studio*, version 10.3.3 Edition développeur).

```
unit HMACO;

interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.UITypes,
  System.Classes, Vcl.Graphics, Vcl.Controls, Vcl.Forms, Vcl.Dialogs,
  MACO, Vcl.StdCtrls, Math, Vcl.ComCtrls, Vcl.Buttons, Vcl.ExtCtrls, Vcl.Menus,
  Vcl.WinXCtrls, Vcl.Samples.Spin, Vcl.Grids, Vcl.Tee.TeeGDIPlus, UTools;

type
  TElement = record
    Bound : Integer;
    Preference : Double;
    Cost : Double;
  end;
  TElements = Array Of TElement;

TFACO = class(TForm)
  Panel4: TPanel;
  Label5: TLabel;
  Label6: TLabel;
  Label7: TLabel;
  StatusBar1: TStatusBar;
  Panel1: TPanel;
  Splitter1: TSplitter;
  Panel3: TPanel;
  Panel2: TPanel;
  B_Save: TBitBtn;
  B_Excel: TBitBtn;
  B_Print: TBitBtn;
  B_Refresh: TBitBtn;
  BitBtn2: TBitBtn;
  BitBtn3: TBitBtn;
  MainMenu1: TMainMenu;
  File1: TMenuItem;
  New1: TMenuItem;
  Open1: TMenuItem;
  Save1: TMenuItem;
  SaveAs1: TMenuItem;
  Print1: TMenuItem;
  PrintSetup1: TMenuItem;
  Exit1: TMenuItem;
  N1: TMenuItem;
  N2: TMenuItem;
  Edit2: TMenuItem;
  Undo1: TMenuItem;
  Repeat1: TMenuItem;
  Cut1: TMenuItem;
  Copy1: TMenuItem;
  Paste1: TMenuItem;
  PasteSpecial1: TMenuItem;
  Find1: TMenuItem;
  Replace1: TMenuItem;
  GoTo1: TMenuItem;
  Links1: TMenuItem;
  Object1: TMenuItem;
  N3: TMenuItem;
  N4: TMenuItem;
  N5: TMenuItem;
  Window1: TMenuItem;
  NewWindow1: TMenuItem;
  Tile1: TMenuItem;
  Cascade1: TMenuItem;
  ArrangeAll1: TMenuItem;
  Hide1: TMenuItem;
  Show1: TMenuItem;
end;
```

```

N6: TMenuItem;
Help1: TMenuItem;
Contents1: TMenuItem;
Index1: TMenuItem;
Commands1: TMenuItem;
Procedures1: TMenuItem;
Keyboard1: TMenuItem;
SearchforHelpOn1: TMenuItem;
Tutorial1: TMenuItem;
HowtoUseHelp1: TMenuItem;
About1: TMenuItem;
SpinEdit1: TSpinEdit;
Label1: TLabel;
Label2: TLabel;
SpinEdit3: TSpinEdit;
Label3: TLabel;
Label4: TLabel;
Label8: TLabel;
Label9: TLabel;
Edit3: TEdit;
Edit4: TEdit;
Edit5: TEdit;
Edit6: TEdit;
ActivityIndicator1: TActivityIndicator;
Label10: TLabel;
Edit7: TEdit;
Label11: TLabel;
Edit8: TEdit;
B_Open: TBitBtn;
Label12: TLabel;
Edit9: TEdit;
Label13: TLabel;
SpinEdit2: TSpinEdit;
Memo1: TMemo;
Splitter2: TSplitter;
Label14: TLabel;
Edit1: TEdit;
OpenDialog1: TOpenDialog;
SaveDialog1: TSaveDialog;
PageControl1: TPageControl;
TabSheet1: TTabSheet;
Memo2: TMemo;
TabSheet2: TTabSheet;
StringGrid1: TStringGrid;
StringGrid2: TStringGrid;
Memo3: TMemo;
procedure BKPACO(Drow:boolean=false);
procedure BitBtn3Click(Sender: TObject);
procedure B_RefreshClick(Sender: TObject);
procedure SaveToFile(FileName:String);
procedure B_SaveClick(Sender: TObject);
procedure LoadFromFile(FileName:String);
procedure B_OpenClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure ReadFromFile(var P : TElements);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  { Private declarations }
  NumberOfSimulations : Integer;
  FeatureSim : DoubleArray;
  IterSim : Integer;
  Projects : TElements;
public
  { Public declarations }
end;

var
  FACO: TFACO;

implementation

{$R *.dfm}

procedure MinMaxArrayNormalize(var A, B : Array of Double; MinValue, MaxValue: Double) ;
var
  I : Integer;

```



```

    L : Double;
begin
    I := Low(A);
    L := MaxValue-MinValue;
    if L=0 then
        while I<=High(A) do
            begin
                B[I]:=MinValue;
                Inc(I);
            end
        else
            while I<=High(A) do
                begin
                    B[I]:=(A[I]-MinValue)/L;
                    Inc(I);
                end;
            end;
        end;

    function MinMaxNormalize(V : Double; MinPref, MaxPref, MinCost, MaxCost: Double):
    Double;
    var
        L : Double;
    begin
        L := MaxPref-MinPref;
        if L=0 then Result := MinCost
        else Result := (V-MinPref)/L*(MaxCost-MinCost)+MinCost;
        end;

procedure TFACO.BitBtn2Click(Sender: TObject);
var
    F : TextFile;
    S : String;
    I: Integer;
begin
    S := '';
    AssignFile(F, ExtractFilePath(Application.ExeName)+'aco.csv');
    Rewrite(F);
    CloseFile(F);
    NumberOfSimulations := StrToInt(Edit1.Text);

    IterSim := 1;
    while IterSim<NumberOfSimulations do
        begin
            BKPACO(False);
            Inc(IterSim);
        end;
        BKPACO(True);
        Memo3.Lines.Clear;
        for I := low(FeatureSim) to High(FeatureSim) do
            S := S+ FloatToStr(FeatureSim[I])+ ' ';
        Memo3.Lines.Add(S)
    end;

procedure TFACO.BitBtn3Click(Sender: TObject);
begin
    Close;
end;

procedure TFACO.B_RefreshClick(Sender: TObject);
begin
    BKPACO(True);
end;
procedure TFACO.BKPACO(Drow : Boolean = False);
var
    minIterations, maxIterations, NumberOfAnts : Integer;
    Alpha, Beta, Q, Ro, TauMax, eps, Budget : Double;
    Ants : TACO;
    Pher : Double2DArray;
    TOF : DoubleArray;
    I, J, Iter, realizable : integer;
    S, Str : String;
    Time1, Time2 : TSystemTime;
    TimeCount : Double;
    F: TextFile;
begin
    AssignFile(F, ExtractFilePath(Application.ExeName)+'aco.csv');
    Append(F);

```

```

minIterations := SpinEdit1.Value;
maxIterations := SpinEdit2.Value;
NumberOfAnts := SpinEdit3.Value;
Alpha := StrToFloat(Edit3.Text);
Beta := StrToFloat(Edit4.Text);
Q := StrToFloat(Edit5.Text);
Ro := StrToFloat(Edit6.Text);
TauMax := StrToFloat(Edit7.Text);
Budget := StrToFloat(Edit8.Text);
eps := StrToFloat(Edit9.Text);
setLength(Pher, 0);
Ants:=TACO.TACO(NumberOfAnts, Alpha, Beta, Q, Ro, TauMax, Budget);
Ants.init;
for I := Low(Projects) to High(Projects) do
  Ants.AddProjects(Projects[I].Cost, MinMaxNormalize(Projects[I].Preference,
    MinPref, MaxPref, MinCost, MaxCost), Projects[I].Bound);

Ants.setAllocatedBudget (StrToFloat(Edit8.Text));

realizable := Ants.isItRealizable;
if realizable = 0 then
begin
  GetLocalTime(Time1);          // store the current date and time

  Iter:=Ants.ACOptimization(minIterations, maxIterations, eps);

  GetLocalTime(Time2);

  S := Ants.getStrResults;

  TimeCount := Time2.wMinute*60+Time2.wSecond+Time2.wMilliseconds/1000-
    Time1.wMinute*60-Time1.wSecond-Time1.wMilliseconds/1000;

  Str :=AddChar(Iter.ToString,6,' ')+';' + Format('%5.3f', [TimeCount])+' ' ;'+
    Ants.getBestPrefer + ' ' ;' + Ants.getBestPlan+
      ' ' ;'+ Ants.getCost+';' +Ants.getBestLength+' ' ;'+S;

  if IterSim=1 then
  begin
    FeatureSim[0]:= TimeCount;
    FeatureSim[1]:= Ants.getBestPrefer.ToExtended;
    FeatureSim[2]:= Ants.getBestPlan.ToExtended;
    FeatureSim[3]:= Ants.getCost.ToExtended;
    FeatureSim[4]:= Ants.getBestLength.ToExtended;
  end
  else
  begin
    FeatureSim[0]:= (FeatureSim[0] * IterSim + TimeCount)/(IterSim+1);
    FeatureSim[1]:= (FeatureSim[1] * IterSim +
Ants.getBestPrefer.ToExtended)/(IterSim+1);
    FeatureSim[2]:= (FeatureSim[2] * IterSim +
Ants.getBestPlan.ToExtended)/(IterSim+1);
    FeatureSim[3]:= (FeatureSim[3] * IterSim + Ants.getCost.ToExtended)/(IterSim+1);
    FeatureSim[4]:= (FeatureSim[4] * IterSim +
Ants.getBestLength.ToExtended)/(IterSim+1);
  end;

  Memo1.Lines.Add(Str);

  Writeln(F, Str);
  if Drow then
  begin
    Pher := Copy(Ants.getPheromones);
    StringGrid1.RowCount := Ants.GetMaxUnits+2;
    StringGrid1.ColCount := Length(Pher)+1;

    for J := 0 to StringGrid1.RowCount do
      StringGrid1.Cells[0,J+1]:= IntToStr(J);
    for I := 1 to StringGrid1.ColCount-1 do
      StringGrid1.Cells[I,0]:= IntToStr(I);

    for J := 1 to StringGrid1.RowCount -1 do
      for I := 1 to StringGrid1.ColCount-1 do
        StringGrid1.Cells[I,J]:= '';

```

```

I:=Low(Pher);
while I <= High(Pher) do
begin
  J := Low(Pher[I]);
  while J<=High(Pher[I]) do
  begin
    StringGrid1.Cells[I+1,J+1] :=Format('%8.3f', [Pher[I,J]]);
    Inc(J);
  end;
  Inc(I);
end;
Memo2.Clear;
TOF := Ants.getTraceObjecFunc;
I:=Low(TOF);
while I <= High(TOF) do
begin
  Memo2.Lines.Add(TOF[I].ToString);
  Inc(I);
end;
end;
end
else
  if realizable = -1 then Memo1.Lines.Add('unrealizable')
else Memo1.Lines.Add('unlimited');

CloseFile(F);
end;

procedure TFACO.B_SaveClick(Sender: TObject);
begin
  SaveToFile('');
end;
procedure TFACO.SaveToFile(FileName:String);
var
  F: TextFile;
  I : Integer;
begin
  if FileName='' then
    if Not SaveDialog1.Execute then Exit
    else FileName:=SaveDialog1.FileName
  else FileName:=ExtractFilePath(Application.ExeName)+FileName;
  AssignFile(F, FileName);
  Rewrite(F);
  Writeln(F, IntToStr(SpinEdit1.Value));
  Writeln(F, IntToStr(SpinEdit2.Value));
  Writeln(F, IntToStr(SpinEdit3.Value));
  Writeln(F, Edit3.Text);
  Writeln(F, Edit4.Text);
  Writeln(F, Edit5.Text);
  Writeln(F, Edit6.Text);
  Writeln(F, Edit7.Text);
  Writeln(F, Edit8.Text);
  Writeln(F, Edit9.Text);
  Writeln(F, Edit1.Text);
  I:= Low(Projects);
  while I<= High(Projects) do
  begin
    Writeln(F, IntToStr(Projects[I].Bound)+Chr(9)+FloatToStr(Projects[I].Preference)
      +Chr(9)+FloatToStr(Projects[I].Cost));
    Inc(I);
  end;
  CloseFile(F);
  MessageDlg('The file' + FileName + ' has been created.',
    mtInformation, [mbOk], 0, mbOk);
end;

procedure TFACO.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  if MessageDlg('Save the work space,mtInformation', mtConfirmation, [mbYes, mbNo,
mbCancel], 0, mbYES)=mrOk then

end;

procedure TFACO.FormCreate(Sender: TObject);
begin
  SetLength(FeatureSim, 5);

```

```

Memol.Lines.Clear;
Memol.Lines.Add(' Iter   Time   Prefer   Cost   Length   Solution (Projects
units''')');
LoadFromFile('aco.par');
end;

procedure SkipChar(Var S : String; var I: Integer; C: Char=' ');
begin
  if I<= length(S) then
    while (I<= Length(S)) and (S[I] in [C, Chr(13), Chr(10)]) do Inc(I);
end;

function ReadNumber(Var Ligne : String; var IndexToken : Integer) : String;
var
  Index : Integer;
  S : String;
begin
  SkipChar(Ligne, IndexToken, ' ');
  S := '';
  Index := IndexToken+1;
  while (Index <= Length(Ligne)) and (Ligne[Index] in ['0'..'9']) do
  begin
    S:=S+Ligne[Index];
    Inc(Index);
  end;
  if Index<Length(Ligne) then
  begin
    If Ligne[Index]='.' then
    begin
      S:=S+'.';
      Inc(Index);
      While (Index <= Length(Ligne)) and (Ligne[Index] in ['0'..'9'])do
      begin
        S:=S+Ligne[Index];
        Inc(Index);
      end;
    end;
    If Ligne[Index]in ['e','E'] then
    begin
      S:=S+'e';
      Inc(Index);
      While (Index <= Length(Ligne)) and (Ligne[Index] in ['0'..'9'])do
      begin
        S:=S+Ligne[Index];
        Inc(Index);
      end;
    end;
  end;
  IndexToken := Index;
  if Length(S)=0 then S:= '0';
  Result:=S;
end;

procedure TFACO.B_OpenClick(Sender: TObject);
begin
  LoadFromFile('');
end;
procedure TFACO.LoadFromFile(FileName:String);
var
  F: TextFile;
  S: String;
  I: Integer;
  Cost, Pref : Double;
begin
  if FileName='' then
    if Not OpenFileDialog1.Execute then Exit
    else FileName:=OpenDialog1.FileName
  else FileName:=ExtractFilePath(Application.ExeName)+FileName;
  AssignFile(F, FileName);
  Reset(F);
  Readln(F, S); SpinEdit1.Value := StrToInt(S);
  Readln(F, S); SpinEdit2.Value := StrToInt(S);
  Readln(F, S); SpinEdit3.Value := StrToInt(S);
  Readln(F, S); Edit3.Text := S;
  Readln(F, S); Edit4.Text := S;
  Readln(F, S); Edit5.Text := S;
  Readln(F, S); Edit6.Text := S;
end;

```

```

Readln(F, S); Edit7.Text := S;
Readln(F, S); Edit8.Text := S;
Readln(F, S); Edit9.Text := S;
Readln(F, S); Edit1.Text := S;
SetLength(Projects, 0);
MinCost := Maxdouble;
MinPref := MinPref;
MaxCost := 0;
MaxPref := MaxCost;
while not Eof(F) do
begin
  Readln(F, S);
  if Length(S) > 0 then
  begin
    I:=0;
    Setlength(Projects, Length(Projects)+1);

    Projects[High(Projects)].Bound:= StrToInt(ReadNumber(S, I));

    Pref := StrToFloat(ReadNumber(S, I));
    Projects[High(Projects)].Preference:= Pref;
    if Pref > MaxPref then MaxPref := Pref;
    if Pref < MinPref then MinPref := Pref;

    {if Cost > MaxCost then MaxCost := Pref;
    if Cost < MinCost then MinCost := Pref;
    }

    Cost := StrToFloat(ReadNumber(S, I));
    Projects[High(Projects)].Cost:= Cost;
    if Cost > MaxCost then MaxCost := Cost;
    if Cost < MinCost then MinCost := Cost;
  end;
end;
CloseFile(F);
StringGrid2.RowCount:=Length(Projects)+1;
StringGrid2.Cells[1,0]:='Bound';
StringGrid2.Cells[2,0]:='Preference';
StringGrid2.Cells[3,0]:='Cost';

I:=Low(Projects);
while I <= High(Projects) do
begin
  StringGrid2.Cells[0,I+1]:='Project '+I.ToString;
  StringGrid2.Cells[1,I+1]:=Projects[I].Bound.ToString;
  StringGrid2.Cells[2,I+1]:=Projects[I].Preference.ToString;
  StringGrid2.Cells[3,I+1]:=Projects[I].Cost.ToString;
  Inc(I);
end;
end;

procedure TFACO.ReadFromFile(var P : TElements);
var
  F: TextFile;
  S: string;
begin
  AssignFile(F, ExtractFilePath(Application.ExeName)+'projects.bkp');
  Reset(F);
  Setlength(P, Length(P)+1);
  Readln(F, S); P[High(P)].Bound:= StrToInt(S);
  Readln(F, S); P[High(P)].Preference:= StrToFloat(S);
  Readln(F, S); P[High(P)].Cost:= StrToFloat(S);
  CloseFile(F);
end;

end.

```

☞ Code source de l'algorithme *génétique* d'optimisation adapté au problème de planification des projets urbains, écrit en langage pascal (Environnement : *Embarcadero RAD Studio*, version 10.3.3 Edition développeur). Cet algorithme est développé pour une comparaison afin d'évaluer l'efficacité de l'algorithme méta-heuristique BKPACS.

```

unit UGAlg;

interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
  System.Classes, Vcl.Graphics, Vcl.Controls, Vcl.Forms, Vcl.Dialogs,
  Vcl.StdCtrls, Vcl.ExtCtrls, Vcl.ComCtrls, Vcl.Menus, Vcl.Samples.Spin,
  Vcl.Buttons, MACO, Math;

type
  TIntArray = Array of Integer;
  TDoubleArray = Array of Double;
  TDNA = class
  public
    Genes : TIntArray;
    fitness : Real;
    planCost : Real;
    constructor Create(targetLength : Word);
    //destructor Destroy; override;
    function GetGenesStr:String;
    Procedure ScoreFitness(var target:TIntArray);
    function genLength:integer;
    function genCost:double;
  end;
  TDNAArray = Array of TDNA;

  TFGAlg = class(TForm)
    Panel1: TPanel;
    Label1: TLabel;
    Edit1: TEdit;
    Memo1: TMemo;
    Panel4: TPanel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    New1: TMenuItem;
    Open1: TMenuItem;
    Save1: TMenuItem;
    SaveAs1: TMenuItem;
    N2: TMenuItem;
    Print1: TMenuItem;
    PrintSetup1: TMenuItem;
    N1: TMenuItem;
    Exit1: TMenuItem;
    MenuItem1: TMenuItem;
    Undo1: TMenuItem;
    Repeat1: TMenuItem;
    N5: TMenuItem;
    Cut1: TMenuItem;
    Copy1: TMenuItem;
    Pastel: TMenuItem;
    PasteSpecial1: TMenuItem;
    N4: TMenuItem;
    Find1: TMenuItem;
    Replacel: TMenuItem;
    GoTo1: TMenuItem;
    N3: TMenuItem;
    Links1: TMenuItem;
    Object1: TMenuItem;
    Window1: TMenuItem;
    NewWindow1: TMenuItem;
    Tile1: TMenuItem;
    Cascadel: TMenuItem;
  end;

```

```

    ArrangeAll1: TMenuItem;
    N6: TMenuItem;
    Hide1: TMenuItem;
    Show1: TMenuItem;
    Help1: TMenuItem;
    Contents1: TMenuItem;
    Index1: TMenuItem;
    Commands1: TMenuItem;
    Procedures1: TMenuItem;
    Keyboard1: TMenuItem;
    SearchforHelpOn1: TMenuItem;
    Tutorial1: TMenuItem;
    HowtoUseHelp1: TMenuItem;
    About1: TMenuItem;
    Timer1: TTimer;
    StatusBar1: TStatusBar;
    Panel2: TPanel;
    B_Save: TBitBtn;
    B_Excel: TBitBtn;
    B_Print: TBitBtn;
    B_Refresh: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    B_Open: TBitBtn;
    Splitter1: TSplitter;
    Panel3: TPanel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    SpinEdit1: TSpinEdit;
    SpinEdit2: TSpinEdit;
    Edit2: TEdit;
    procedure AddProject(w, v : double; u: integer);
    function GA(Var population : TDNAArray; var target : TIntArray;
        totalPopulation : Integer; mutationRate : Real; popMax : Integer):double;
    procedure B_RefreshClick(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);

private
    { Déclarations privées }
public
    { Déclarations publiques }
end;

var
    FGAlg: TFGAlg;

implementation

{$R *.dfm}

var
    projects : TIntArray;
    cost : TDoubleArray;
    profit : TDoubleArray;
    MaxCost : Double;

    Attributes : TDoubleArray;
    population : TDNAArray;

function RandomUnits(var I : Integer):Integer;
begin
    Result := random(projects[I]+1);
end;

constructor TDNA.Create(targetLength : Word);
Var
    I : Integer;
begin
    SetLength(Genes, targetLength);
    I:=Low(Genes);
    While I <= High(Genes) do
    begin
        Genes[I]:= RandomUnits(I);
        Inc(I);
    end;
end;

```

```

end;

function TDNA.genLength:integer;
begin
  Result := Length(Genes);
end;

function TDNA.genCost;
var
  I : Integer;
  Tot: double;
begin
  Tot:=0;
  I := Low(Genes);
  while I <= High(Genes) do
  begin
    Tot:=Tot+Genes[I]*cost[I];
    Inc(I);
  end;
  result := Tot;
end;

function GetDNA(var DNA:TDNA): TDNA;
var
  Res : TDNA;
  I : Integer;
begin
  Res:=TDNA.Create(length(DNA.Genes));
  I:=Low(DNA.Genes);
  While I <= High(DNA.Genes) do
  begin
    Res.Genes[I]:=DNA.Genes[I];
    Inc(I);
  end;
  Res.fitness:=DNA.fitness;
  Result := Res;
end;

function crossover(var partnerA, PartnerB : TDNA):TDNA;
var
  child : TDNA;
  I, genLength, midPoint : Word;
begin
  genLength:=partnerA.genLength;
  child:=TDNA.Create(genLength);
  midPoint:=Random(genLength-1);
  I :=0;
  While I < midpoint do
  begin
    child.Genes[I]:=partnerA.Genes[I];
    Inc(I);
  end;
  While I < genLength do
  begin
    child.Genes[I]:=partnerB.Genes[I];
    Inc(I);
  end;
  Result:=child;
end;

procedure mutate(Var DNA: TDNA; mutationRate: Real);
Var I : Integer;
begin
  I:= 0;
  while I < DNA.genLength do
  begin
    if Random < mutationRate Then
      DNA.Genes[I]:=RandomUnits(I);
      Inc(I);
    end;
  end;
end;

function TDNA.GetGenesStr:String;
Var
  I : Word;
  S : String;
begin

```



```

S:='';
I:=Low(Genes);
While I <= High(Genes) do
begin
  S:=S+IntToStr(Genes[I])+' ';
  Inc(I);
end;
Result := S;
end;

procedure TDNA.ScoreFitness(var target: TIntArray);
Var
  I : Integer;
  Score1, Score2, Score3, C : Double;
begin
  Score1 := 0;
  Score2 := 0;
  Score3 := 0;
  C:=0;
  I:=Low(Genes);
  While I <= High(Genes) do
  begin
    Score1 := Score1 + Genes[I]*Profit[I];
    Score2 := Score2 + Genes[I]*Cost[I];
    //Score3 := Score3 + Genes[I]*Profit[I];
    C:=C+Profit[I]*Target[I];
    Inc(I);
  end;
  Score1 := Score1/C;
  Score2 := Score2/MaxCost; if Score2>1 then Score2:=1/(Score2*Score2);
  //Score3:=Score3/C;
  fitness := (score1*Score2);
end;
{~~~~~}
{Initialize population. the required parameters are the number of initial
{population and lengthe of the target
{~~~~~}

procedure Setup(Var Population : TDNAArray; numberOfInitialPopulation,
  targetLength : Word);
var
  I : Word;
begin
  SetLength(Population, numberOfInitialPopulation);
  I:= Low(Population);
  While I <=High(Population) do
  begin
    Population[I]:=TDNA.Create(targetLength);
    Inc(I);
  end;
end;

{~~~~~}
{Selection
{~~~~~}

function Selection(Var population : TDNAArray; Var matingPool : TIntArray;
  var target: TIntArray; mutationRate: Real): Integer;
Var
  I, J : Integer;
  maxScore : Real;
begin
  MaxScore := 0;
  J:=0;
  SetLength(matingPool, Length(Population));
  I:= Low(Population);
  While I <=High(Population) do
  begin
    Population[I].ScoreFitness(target);
    if Population[I].fitness>maxScore then
    begin
      maxScore := Population[I].fitness;
      J:=I;
    end;
    matingPool[I] := Trunc(population[I].fitness*100);
    //Add each member n times according to
  if I>0 then matingPool[I]:= matingPool[I]+matingPool[I-1];
  Inc(I);

```

```

    end;
    Result := J;
end;

function SeekIndividu(var matingPool : TIntArray; k : Integer): Integer;
var
    I : Integer;
    Tr : Boolean;
begin
    Tr := False;
    I := low(matingPool);
    while (I <= High(matingPool)) and Not Tr do
    begin
        if matingPool[I]>=k then Tr := True
        else Inc(I);
        end;
    Result := I;
end;

Procedure Reproduction(Var population : TDNAArray; Var matingPool : TIntArray;
    mutationRate: Real);
Var
    I, A, B : Integer;
    partnerA, PartnerB, child : TDNA;
    newPopulation : TDNAArray;
begin
    SetLength(newPopulation, Length(Population));
    I:=Low(Population);
    While I <= High(Population) do
    begin
        A:=Random(matingPool[High(matingPool)]);
        B:=Random(matingPool[High(matingPool)]);
        partnerA:=getDNA(population[SeekIndividu(matingPool, A)]);
        partnerB:=getDNA(population[SeekIndividu(matingPool, B)]);
        child := crossover(partnerA, partnerB);
        mutate(child,mutationRate);
        newpopulation[I]:=getDNA(child);
        Inc(I);
    end;
    Population:=Copy(newPopulation);
end;

function TFGAlg.GA(Var population : TDNAArray; var target : TIntArray;
    totalPopulation :Integer; mutationRate : Real; popMax : Integer):double;
var
    J : Integer;
    matingPool : TIntArray;
    bestScore : Real;
    Generations : Integer;
begin
    Generations:=0;
    Randomize;
    setup(population, TotalPopulation, Length(target));
    Repeat
        J:= Selection(Population, matingPool, target, mutationRate);
        bestScore := Population[J].fitness;

        //Memol.Lines.Add(population[J].GetGenesStr+'-->Cost = '+
    population[J].genCost.ToString);
        //if (Memol.Lines.Count mod 177 = 0) then Memol.Clear;
        Reproduction(Population, matingPool, mutationRate);
        Inc(Generations);
    Until (BestScore >= 1-mutationRate) or (Generations>=popMax);
    Label1.Caption:=IntToStr(Generations);
    Memol.Lines.Add(population[J].GetGenesStr+'-->Cost = '+ population[J].genCost.ToString);
    Result := population[J].genCost;
end;

procedure TFGAlg.BitBtn3Click(Sender: TObject);
begin
    Close;
end;

procedure TFGAlg.AddProject(w, v : double; u: integer);
begin
    setlength(projects, length(projects)+1);
    setlength(cost, Length(Projects));
    setlength(profit, Length(Projects));

```

```

    projects[high(Projects)]:= U; cost [high(cost)]:= w; profit[high(profit)]:=v;
end;

procedure TFGAlg.B_RefreshClick(Sender: TObject);
var
    I: Integer;
    Tot:double;
begin
    Memol.Lines.Clear;
    Edit1.Text:='';
    setlength(projects, 0);
    setlength(cost, Length(Projects));
    setlength(profit, Length(Projects));
    AddProject(9,150, 1);
    AddProject(13, 35, 1);
        AddProject(153,200,2);
        AddProject(50,60,2);
        AddProject(15,60,2);
        AddProject(68,45,3);
        AddProject(27,60,3);
        AddProject(39,40,3);
        AddProject(23,30,1);
        AddProject(52,10,3);
        AddProject(11,70,1);
        AddProject(32,30,1);
        AddProject(24,15,2);
        AddProject(48,10,2);
        AddProject(73,40,1);
        AddProject(42,70,1);
        AddProject(43,75,1);
        AddProject(22,80,1);
        AddProject(7,20,1);
        AddProject(18,12,2);
        AddProject(4,50,1);
        AddProject(30,10,2);

    MaxCost := 400;
    for I := Low(Projects) to High(Projects) do
    begin
        Edit1.Text :=
Edit1.Text+'('+projects[I].ToString+', '+cost[I].ToString+', '+profit[I].ToString+') ';
    end;
    Edit1.Text:=Edit1.Text+'Max Cost = '+maxCost.ToString;

    Tot:=GA(population, projects, SpinEdit2.Value, StrToFloat(Edit2.Text),
        SpinEdit1.Value);
    Memol.Lines.Add(Tot.ToString);
end;

end.

```