Université Fédérale

Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Université Toulouse - Jean Jaurès

Présentée et soutenue par :
Sami MNASRI
le mercredi 27 Juin 2018

Titre :

Contributions to the optimized deployment of connected sensors on the Internet of Things collection networks

---

École doctorale et discipline ou spécialité :

ED MITT : Domaine STIC : Réseaux, Télécoms, Systèmes et Architecture

Unité de recherche :

Institut de Recherche en Informatique de Toulouse (IRIT), équipe RMESS

Directeur de Thèse :

Thierry VAL, Professeur à l'Université de Toulouse

Jury :

Pr. Belhassen ZOUARI, Université de Carthage, Rapporteur

HDR Dr. Hanen IDOUDI, Ecole Nationale des Sciences de l'nformatique de Tunis, Rapporteur

Dr. Najeh NASRI, Ecole Nationale d'Ingénieur de Sfax, Co-Directeur de thèse

HDR Dr. Adrien VAN DEN BOSSCHE, l'Université de Toulouse, Co-Encadrant de thèse

Nothing is stronger than a high-quality education system that can be the basis of building human being in order to reform morals towards modern civilizations and nations

Sami Mnasri

To my mother and my father,
To my sisters and my brother,
To Ameni,
To my friends,

In testimony of my great love
and my strong gratitude
Sami

Acknowledgements

I would first like to thank Pr. Thierry VAL, my thesis director for his support and contribution in this research as well as for his availability. In particular, I thank him for giving me a great freedom in my research, for being attentive to all my comments or ideas and for answering my many questions. His encouragement and advices were invaluable, without which this work would probably not have been possible.

I thank Dr Nejah NASRI for his interest in my work as well as for his comments and advices which allowed me to improve the substance and form. I would like to thank Dr. Adrien VAN DEN BOSSCHE for his advice and guidelines, especially at the level of experiments and prototyping that have enhanced my work.

I also thank Pr. Belhassen ZOUARI and Dr. Hanen IDOUDI for agreeing to be part of my evaluation committee.

My thanks also go to all the teachers, staff and my students on the IUT of Blagnac for the good atmosphere they created in the premises of the IUT.

Finally, I thank all the teachers who contributed to my training and to all those who helped me during this work.

About this dissertation

IoT collection networks raise many optimization problems, in particular because the sensors have limited capacity in energy, processing and memory. We are interested in a global contribution related to the optimization on wireless sensor networks using heuristics, meta-heuristics, hybrid methods, mathematical models, assessments of new optimization approaches on wireless sensor networks, multi-objective models, etc.
The continuous evolution of the IoT collection network research domain and technology allows optimizing the use of these networks in different number of contexts. The general problem of deployment of nodes can be described as follows: often, the sensors constituting the network cannot be precisely positioned and are scattered erratically. In order to compensate the randomness of sensor placement and increase the fault tolerance of the network, these sensors can be deployed intelligently. Unlike 3D deployment, there is a very abundant literature on this problem for 2D deployment ranging from the technical specificities of the sensors and their way of communicating, to the topological organization of the network itself. The 3D deployment of wireless sensors poses many optimization problems. Therefore, the goal is to provide 3D sensor organization solutions and find the most optimized architecture in order to improve the network performance.
In this context, heuristic approaches for optimization in wireless sensor networks can be envisaged. In addition, optimization problems are generally derived from real problems that are often antagonistic, unmeasurable and include several criteria. The dynamic, distributed and open aspect of the deployment problem led us to adopt advanced modeling and optimization technologies: hybridizing recent meta-heuristics. Indeed, in order to ensure a continuous evolution, a pragmatic flexibility and an infallible robustness against possible disturbances that could affect all or part of the network, the main goal was to propose hybridizations and modifications of the evolutionary optimization algorithms in order to achieve the positioning of nodes in wireless sensor networks with satisfaction of a set of constraints and objectives. In fact, hybrid meta-heuristics exploit the complementarity of these methods with each other, as well as with other "classical" approaches. This new class of hybrid algorithms has proven its robustness and efficiency in solving difficult optimization problems. We propose to focus our contribution composed of several complementary and progressive approaches, on the hybrid meta-heuristics. Especially on the evaluation of their performances and their applications on our real-world problem. These hybridization schemes are all validated by numerical results by the evaluation of algorithms with metrics such as the Hypervolume. Then, simulations supplemented by real experiments on testbeds were proposed. Finally, simulations are confronted with experiments to evaluate the behavior of algorithms and prove their stability and efficiency.

Keywords: 3D indoor deployment, ant colony algorithm, dimensionality reduction, experimental validation, genetic algorithms, hybridization, IoT collection networks, meta-heuristics, optimization, particle swarm optimization, user preferences.

Contents

List of figures

List of algorithms

List of tables

List of acronyms

acMaPSO	accent-based Many-objective Particle Swarm Optimization
acMaMaPSO	accent-based many-objective multi-agent Particle Swarm Optimization
AcNSGA-III	Ant colony NSGA-III
ACO	Ant colony optimization
AGP	Art Gallery Problem
AHCH	Adaptive Hole Connected Healing
AI	Artificial Intelligence
AIS	Artificial Immune System
AmN	Adaptive multi-Mutation strategy with Neighbourhood restrictions
AODV	Ad-hoc On-demand Distance Vector
AxN	Adaptive multi-Recombination with Neighbourhood restrictions
BCO	Bees Colonies Optimization
BDA	Bernoulli Deployment Algorithm
BDI	Belief-Desire-Intention (model)
BS	Base Station
CA	Convergence Archive
cNSGA-III	constrained NSGA-III
CSMA/CA	Carrier-sense multiple access with collision avoidance
DA	Diversity Archive
DAI	Distributed Artificial Intelligence
dBm	Decibel-milliwatt (unit)
DDA	Differentiated Deployment Algorithm
DL-IoT	Device Layer -Internet of Things
DM	Decision Maker
DRP	Dominance Relation Preservation-based
DSN	Directional Sensor Networks
DTLZ	Deb-Thiele-Laumanns-Zitzler (test problems)
DV-Hop	Distance Vector Hop
EA	Evolutionary Algorithm
EC-NSGA-II	Extremized-Crowded NSGA-II
EMaO	Evolutionary Many-objective Optimization
EMO	Evolutionary Multi-objective Optimization
EMOA	Evolutionary Multi-objective Optimization Algorithm
FER	Frame Error Rate
FS	Feature Selection
GD	Generational Distance
GNVG	Generational Non-dominated Vector Generation
GNVGR	Generational Non-dominated Vector Generation Ratio
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HV	HyperVolume
IGD	Inverted Generational Distance
IoT	Internet of Things
JADE	Java Agent DEvelopment framework
LoS	Line of sight
LQI	link quality indication
LoRa	Long Range
LS	Local Search
L-PCA	Linear Principal Component Analysis
MaOA	Many-objective optimization Algorithm
MaOEA	Many-objective Optimization Evolutionnary Algorithm
MaOP	Many-objective Optimization Problem
MaOPSO	Many-objective Particle Swarm Optimization
MAS	multi-agent systems

MCDM	Multi-Criteria Decision Making
MOEA	Multi-objective Optimization Evolutionary Algorithm
MOEA/D	Multi-objective Evolutionnary Algorithm based on Decomposition
MOEA/DD	Multi-objective Evolutionnary Algorithm based on Decomposition and Dominance
MOGA	Multi-Objective Genetic Algorithm
MOP	Multi-objective Optimization Problem
MOPSO	Multi-Objective Particle Swarm Optimization
MVU	Maximum Variance Unfolding
NL-MVU-PCA	Non-Linear-Maximum Variance Unfolding-Principal Component Analysis
NPGA	Niched Pareto Genetic Algorithm
NSGA	Non-dominated Sorting Genetic Algorithm
NU	Network Utilization
NVA	Non-dominated Vector Additional
OMNeT++	Objective Modular Network++
ONVG	Overall Non-dominated Vector Generation
ONVGR	Overall Non-dominated Vector Generation Ratio
PAES	Pareto Archived Evolutionary Strategy
PBI	Penality-based Boundary Intersection
PCA	Principal Component Analysis
PCS	Pareto Corner Search based
PCSEA	Pareto corner search evolutionary algorithm
PESA	Pareto Envelope-based Selection Algorithm
PF	Pareto front
PFDA	Potential Field Deployment Algorithm
PI-EMOA	Progressively Interactive EMO algorithm
PI-EMO-PC	Progressive-Interactive-Evolutionary Multi-objective Optimization-Polyhedral Cone
PI-EMO-PC-INK	Progressive-Interactive-EMO- Polyhedral Cone-Ideal point-Nadir point-Knee points
PI-EMO-VF	Progressive-Interactive-Evolutionary Multi-objective Optimization-Value-Function
PL	Path loss
PP	Positioning Problem
PSO	Particle Swarm Optimization
RCM	Reduced Correlation Matrix
RERR	Route Error
RoI	Region Of Interest
RREP	Route Reply
RREQ	Route Request
RSSI	Received Signal Strengh Indication
SA	Simulated Annealing
SBX	Simulated Binary Crossover
SMCP	Set MultiCover Problem
SOP	Single-objective Optimization Problem
SPEA	Strength Pareto Evolutionary Approach
TCB	Weighted Tchebycheff Distance
TKR-NSGA-II	Trade-off-based KR-NSGA-II
TS	Taboo Search
Two_Arch2	Two_Arch2 Two-Archive2 (algorithm)
UASN	Underwater Acoustic Sensor Network
UFS	Unsupervised Feature Selection-based
USN	Underwater Sensor Network
UWB	Ultra Wide Band
U-NSGA-III	Unified NSGA-III
VEGA	Vector Evaluated Genetic Algorithm
VFA	Virtual Force Algorithm
VP	Voronoi Partition
WFG	Walking Fish Group (test problems)
WiFi	Wireless Fidelity
WiNo	Wireless Node
WS	Weighted Sum
WSN	Wireless Sensor Network
ZDT	Zitzler-Deb-Thiele (test problems)

Introduction and overview

Motivations and problematic

With different contexts of application, wireless sensor networks (WSN) are a research field that is in continuous evolution, especially with the emergence of the Internet of Things (IoT). Indeed, IoT is a concept that is closely related to the issues discussed in our study. IoT is a scenario in which different heterogeneous and communicating entities called objects or things (people, robots or devices) are connected and distinguished by unique identifiers. These entities can automatically transfer data to the network without any human intervention. WSN is the bridge connecting the real world to the digital one. It provides hardware communication to transmit and retrieve real values detected by wireless connected objects (sensor nodes). While the role of IoT is to process this data, manipulate it and make the decisions. In this thesis, we are interested in DL-IoT (DeviceLayer-IoT) networks which are collection networks used to collect data from distributed sensor nodes in the environment of the network. Hence, our approach can be applied for both WSN and IoT contexts. In this respect, given the limited energy, processing and memory capacity of sensors/objects, DL-IoT collection networks raise many optimization problems. Our global contribution is the proposal of heuristics, hybrid meta-heuristics (centralized and distributed), multi-objective models, mathematical formulations and evaluations of recent optimization approaches on DL-IoT collection networks. Given the continuous growth in the themes of DL-IoT collection networks, different contexts can be reversed such as optimizing the deployment of nodes.

Deploying the nodes is the first step in installing a WSN. In terms of energy consumption optimization, this first step greatly affects the performance, reliability and operation of the network. The problem of deploying the nodes can be described as the positioning of all the sensors (that can be randomly scattered initially) constituting the network. To compensate their erratic nature of placement, a large number of sensors are often deployed intelligently. This can also contribute in increasing the fault tolerance of the network. For the 2D deployment of WSN, a very abundant literature exists ranging from the topological organization of the network to the technical specificities of the sensors and their way of communicating. However, the low-cost 3D deployment poses many optimization problems with a limited literature. Hence, our goal is to find the most optimized architecture and provide 3D sensor solutions, while improving the network performance. In this respect, several issues and objectives are related to the problem of deployment of sensor networks such as energy consumption, lifetime and localization. As a result, different heuristic approaches can be considered for optimizing WSNs.

The distributed and dynamic nature of the deployment problem requires the use of advanced and recent optimization and modeling methodologies: hybrid meta-heuristics. These latters allow exploiting the complementarity of these methods with each other and taking advantage from the benefits of other conventional approaches hybridized with them. This new class of hybrid meta-heuristics has shown its performance in solving difficult optimization problems especially when faced by objectives (like ours) that mainly concern the assurance of an evolutionary, flexible and robust system against network disturbances. Our contribution consists in putting a set of well justified modifications and hybridizations on the meta-heuristics, then applying them and evaluating their performances on our real world problem of deployment. In this respect, the most difficult optimization problems to be solved are generally derived from real problems that are often unmeasurable, complex and antagonistic.

In general, these problems include several criteria. This is due to the fact that they have several evaluation objectives; often contradictory; to be considered simultaneously. This gave birth to the theory of multi-objective optimization. In the literature, several methods of solving multi-objective problems have been developed. These methods can be classified into two main classes: exact methods and approximate ones that are subdivided into meta-heuristics and specific heuristics. Meta-heuristics form a set of optimization algorithms that aim to solve difficult optimization problems. These algorithms allows improving the quality of the solutions without guaranteeing the optimality of the obtained solution but in a very reasonable calculation time with respect to the complexity of the problem, often unsolvable in a polynomial time. Real-world optimization problems often have several conflicting and contradictory objectives and constraints (called multi-objective optimization problems (MOP)). This implies that there is not a single solution that is optimal in relation to all these objectives in the same time. The output of such multi-objective optimization algorithms is generally composed of a set of incomparable non-dominated solutions. These solutions are called "Pareto front" (PF). The goal of multi-objective optimization is to find a well-distributed and well-converged approximation of the PF. Subsequently, the decision maker (DM) can select the preferred solution. To determine the PF of a problem, various methods that are based on the imitation of the principles of biological evolution, have been proposed in the literature. These methods, named evolutionary algorithms (EA), have become popular and widely used in the resolution of MOPs because of their insensitivity to the objective function aspects and forms such as multimodality, discontinuity, convexity and its search space uniformity (Deb 2001). Indeed, evolutionary multi-objective optimization (EMO) is a new branch of optimization that has emerged following this success of the Multi-Objective Optimization EA (MOEA) on MOP resolving.

Research aims and principal contributions

The aim of this thesis is to propose hybridizations and modifications of evolutionary optimization algorithms in order to realize the adequate positioning of nodes in WSN while satisfying a set of constraints and objectives.

- As a first stage, an in-depth literature review was conducted on the methods of optimizing the deployment in WSN, especially the methods resolving the 3D indoor deployment. This study covers swarm-based meta-heuristics (particle swarm optimization, ant colony optimization and bee hives algorithm), genetic algorithms, taboo research and simulated annealing. The study also deals with the single-objective and multi-objective case; the static and dynamic case; the distributed and parallel case of deployment.
- In the second stage, a mathematical formulation based on an Integer Linear Programming model and aims at modeling the problematic is proposed. This formulation identifies and explains the objective function to optimize, the decision variables and the different constraints to be taken into consideration. Our goal is to minimize the number of sensor nodes to use and the energy consumption. At the same time, maximizing the network lifetime, the coverage, the localization and connectivity.
- In the third stage, the focus is on the proposition of different justified hybridization and modifications schemes introduced on the optimization algorithms in order to better solve the deployment problem:
  - In a first approach, an adaptive mutation and recombination operators with neighborhood mating constraints are proposed. The use of a concept of multiple scalarization functions is introduced to deal with: (i) the inefficiency of Pareto-based multi-objective evolutionary algorithms, (ii) the inefficiency of the

recombination operation, and (iii) the exponential increase in costs (time and space) when solving multi-objective problems in the real world context.

- In a second approach, the incorporation of preferences is established: In the case of real world problems having a large number of objectives, the size of the population and the number of necessary solutions depend exponentially on number of objectives. Thus, the performance of the optimization algorithms deteriorates when solving such problems. To solve this challenge, the NSGA-III algorithm is hybridized with an interactive user preference strategy (PI-EMO-VF) that follows the evolution of new solutions to gradually incorporate the preferences of the user. In this work, the effectiveness of the NSGA-III is tested in real-world problems, and compared to another recent multi-objective algorithm (MOEA/DD).
- In a third approach, a justified hybridization scheme that combines three classes of multi-objective algorithms. These algorithms rely on reference points (NSGA-III, MOEA/DD), aggregation (Two_Arch2) and decomposition (MOEA/D) with two procedures based on dimensionality reduction (MVU-PCA) and preferences (PI-EMO-PC). The purpose of this hybridization is to benefit from the advantages of each method to solve our complex problem.
- In a fourth approach, a new hybrid algorithm derived from biological observations (ant search behavior and genetics) is proposed. It is based on the recent variant of the genetic algorithms (NSGA-III) and the ant colony algorithm (ACO). This is the first time NSGA-III and ACO are integrated into a hybrid platform. Moreover, unlike traditional hybridizations, these two algorithms iterate at the same time and interact using the same population (the initial population of the NSGA-III is the population built by the ants in the initial phase of the ACO in the same iteration). Then, the steps of the ant algorithm are injected into the NSGA-III with incorporation of several modifications on the original NSGA-III.
- In a fifth approach, a particle swarm optimization algorithm (called acMaPSO) based on a new bird accent concept is proposed. The new bird accent concept relies on a set of birds that are separated into different accent groups by their regional dwelling and are classified into groups according to their ways of singing. This new concept of bird accent is introduced to preserve the diversity of the population during research and to assess the particle search capability in their local areas. To ensure that the search escapes local optima, the most expert particles (the parents) "die" and are regularly replaced by new particles that are randomly generated.

  It is also proposed to test this hybridization in a distributed environment. For this purpose, it is proposed to hybridize the acMaPSO algorithm with a multi-agent system (MAS). The new variant (named acMaMaPSO) takes advantage of agent distribution and particle interactivity. We propose a decentralized multi-agent architecture that contains three types of agents: an environment agent, swarm agents and bird agents (or particle agents). These agents have different knowledge, goals, abilities and action plans.

All these proposed hybridizations are tested and approved by numerical results that use evaluation metrics such as the Hypervolume. Afterwards, simulations and prototypes on real testbeds have been proposed. Then, in order to prove the stability and efficiency of these hybridizations in different contexts, the simulations are confronted with real experiments.

Structuring of the document

This document is structured into four chapters divided in two parts. The first part illustrates the state of the art in the first two chapters; and the second part details the proposed contributions in chapters 3 and 4.

The first chapter starts by presenting the state of the art of the 3D indoor deployment problem. We introduce the issues of the three-dimensional deployment and its different types, objectives, models and applications. Then we identified and criticized the recent research works dealing with the problem of deployment.

The second chapter introduces the methods used to solve the 3D indoor deployment, especially the evolutionary optimization algorithms, the multi-agent systems and the incorporation of dimensionality reduction and user preferences. We also detail the fundamental concepts of multi-objective optimization such as the dominance and the Pareto front. Then we describe the main classical approaches and meta-heuristic of resolution of multi-objective problems.

In the third chapter is dedicated to illustrate our mathematical modeling of the deployment problem based on an integer linear programming formulation. We present the architecture of nodes, the assumptions, the notation and the objective function. Then we detail the considered objectives and their specificities. Moreover, this chapter is devoted to explain and justify the proposed hybridizations and modifications that are introduced to meta-heuristics to improve their performances and capabilities to solve optimization problems. Especially complex and many-objective real world problems like ours. These changes relies essentially on the following contributions: the use of the neighborhood and adaptive recombination operators; the use of multiple scalarizing functions in the aggregation based approach; the incorporation of dimensionality reduction and users preferences; a hybrid framework for NSGA -III and Ant system; and finally the proposition of a concept of bird accents in PSO and its hybridization with MAS.

The last fourth chapter is devoted to illustrate the results of various evaluations. We initially introduced the different test parameters and evaluation metrics that we used to validate our proposals. We turn next to illustrate the numerical results of the 3D deployment problem after incorporating our proposed hybrid scheme of dimensionality reduction and preferences. Then we present the results on the DTLZ unconstrained test functions. Afterward, we propose a study that investigates the modeling of the used protocols and presents the simulations with small and large instances. Next, we propose ACO and NSGA-III experiments, then the results of the based-on bird's accent PSO algorithm and the PSO-MAS one.

Finally, we present our conclusions as well as our perspectives and futures directions of research.

Part I
State of the Art

Chapter 1
The 3D indoor deployment problem in WSN and DL-IoT collection networks

1.1 Introduction

Deployment represents a fundamental role in setting up efficient wireless sensor networks (WSN). In general, WSN are widely used in a variety of applications ranging from monitoring a smart house (planned deployment) to monitoring forest fires with parachuted sensors (random deployment). In this first chapter, we focus on the planned deployment, in which the sensor nodes must be accurately positioned at predetermined locations to optimize one or more design objectives of the WSN, under some given constraints. The purpose of planned deployment is to determine the type, number, and locations of nodes to optimize coverage, connectivity and network lifetime. There have been a large number of studies, which proposed algorithms for solving the premeditated deployment problem. The main purposes of this chapter are two-fold. The first one is to present the complexity of 3D deployment and then detail the types of sensors, objectives, applications and recent research that concerns the strategy used to solve this problem. The second one is to present a comparative survey between recent optimization approaches used to resolve the deployment problem in WSN. Based on our extensive review, we discuss the strengths and limitations of each proposed solutions and compare them in terms of the different WSN design factors.

1.2 Presentation, migration from WSN to DL-IoT and complexity of the deployment problem

Another concept is closely related to our problem, it is the IoT (Internet of Things). The IoT is a scenario in which entities (devices, people or robots) are connected and have a unique identifier for each. They are able to transfer data over the network without human or automatic intervention. These objects communicate using protocols such as the Bluetooth or the 802.15.4 as it is the case in our experiments (section 4.5). WSN is the bridge which links the real world to the digital one. It is responsible for the hardware communication to convey the real world values detected by the wireless connected things (sensor nodes) to the Internet. While the IoT is responsible for data processing, manipulation and decision making. In our study we are interested in the DL-IoT (DeviceLayer-IoT) (Van den Bossche et al, 2016) which is a network of collection used to collect data from the sensors nodes disseminated in the network environment. Thus, our approach and contributions are valid for both WSN and IoT contexts.

Indeed, a fundamental problem in the WSN is how to deploy sensor nodes. The deployment problem can be described as follows: Having N wireless sensors and an area A to cover, how to deploy these sensors to form a WSN that meets system requirements, such as the connectivity of the network, its ability to detect relevant events happening in A, and its ability to provide a required period of operation. This problem is related to coverage, connectivity, and lifetime issues. Indeed, the problem of deployment in WSN refers to the determination of the positions of nodes (and/or the base stations) so that the coverage, the connectivity and the energy efficiency can be obtained with a minimum number of nodes.

Events happening in an area lacking a sufficient number of nodes may be unnoticed. Moreover, areas with dense sensor populations suffer from congestion, redundancy detection, and delays. Optimal deployment ensures adequate quality of service, long network life and cost saving. The problem of deployment can also be defined as follows: having a surface A in 2D or 3D with a set of obstacles (if these obstacles exist, they should not partitionate A) and a set of sensors with different types (according to their radii of detection and communication), the overall goal is to minimize the number of nodes to be deployed on A while ensuring one or more objectives such as network coverage and connectivity. It is said that the target region is covered if each point in A is covered by an active sensor having a probability of coverage Pc and a sensing range R_s if there is direct communication (Line of Sight). Besides, it is said

that the network monitoring a target region is *fully connected* if all sensors can route data in multi-hop to the base station or to another destination node.

Most deployment approaches consider a WSN with randomly deployed sensors (Ammari, 2014) which are generally modeled by a Poisson point process with a density γ. This Poisson process is defined as follows: for any zone A in the region R, the distribution of the number of nodes in A is the mean distribution of Poisson $\gamma\|A\|$, $\|A\|$ is the surface of A (Ammari, 2014). Given the number of nodes, their locations are mutually independent random variables and uniformly distributed over A. It is known that n nodes whose locations are mutually independent random variables, with a uniform distribution within A, are mainly associated to a Poisson process with a density γ, provided that A is large (Hall, 1988).

An example of randomly deployed sensor networks is shown in Figure 1.1. There are twelve sensors deployed in a rectangular detection field. The density of sensors on the left side of the field is higher than that on the right one. Therefore, the detection field is not fully covered by the sensors.

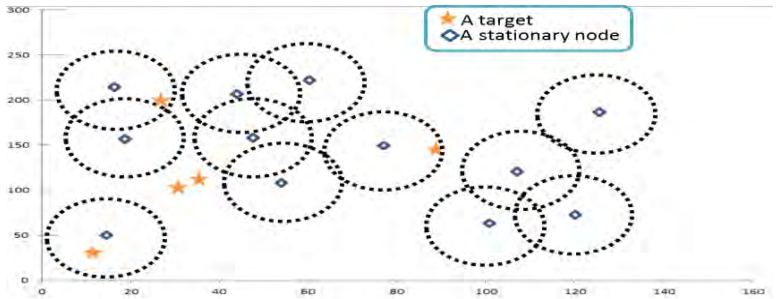
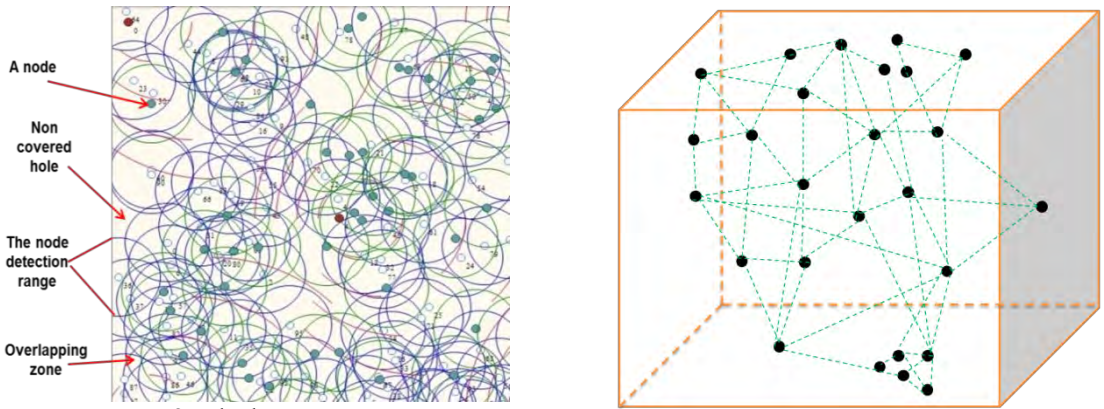


Figure 1.1 An example of area coverage using randomly deployed sensors

Finding the optimal node distribution is a difficult problem to solve and is considered NP-hard for most formulations. This problem is proven to be optimally solved in 2D environments while it has been proven to be NP-difficult if it is generalized to 3D environments (Cheng et al., 2008). Figure 1.2 shows an example of 2D deployment and another for 3D deployment.



2D deployment 3D deployment

Figure 1.2 The 2D deployment vs. the 3D deployment

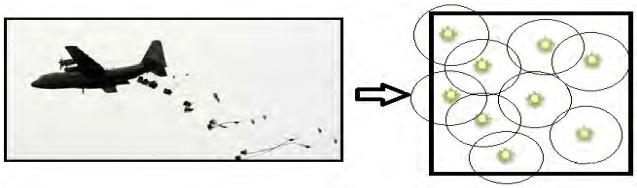
When sensors are deployed randomly, the initial coverage area provided by the network cannot be optimal as in the case of deterministic deployment. In order to increase the covered area, redundant sensors are deployed. Redundancy makes sensor networks denser than normal ad hoc networks. However, increasing sensor density cannot provide a 100% coverage probability. In addition, it is expensive to maintain high-density WSN on a large scale. Therefore, other approaches should be used to avoid these problems and improve the

coverage after the initial random deployment. Another problem that complicates the redeployment is that of the robustness of WSN. Indeed, once deployed, it is expensive and impractical, even impossible, to replace unusable sensors in most types of applications. Hence, if a particular node is no longer running, there will be an impact on the overall performance of the network. The loss of a node may be due to different causes, such as battery depletion, physical damage caused by environmental forces, or destruction by the enemy. If a sensor that covers a sensitive area dies and no other sensor can cover that area, the WSN fails its mission of efficiently distributing the sensors.

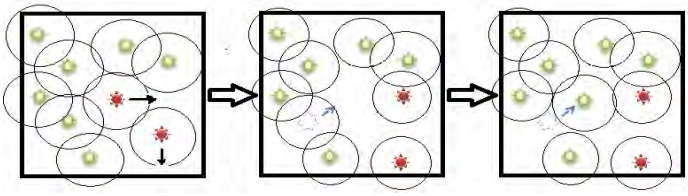
Indeed, one way to optimize the distribution strategy in WSN is to have redundant sensors that improve the performance of the network. In addition, if the detection field is vast or has limited access, the sensors may not be able to be deployed one by one in specific locations. Instead, they can be disseminated from an airplane. When the sensors are randomly deployed, the initial coverage area provided by the network cannot be optimal as in the case of deterministic deployment. In order to increase the coverage area, redundant sensors can be deployed. Redundancy makes sensor networks denser than ad hoc networks. However, increasing sensor density can not provide a 100% coverage probability. Even more, it is expensive to maintain high-density sensor arrays on a large scale. Therefore, other approaches should be used in order to avoid these problems and improve the coverage after the initial random deployment. Another problem that complicates the redeployment is the robustness of WSN. Once deployed, it is expensive and impractical, if not impossible, to replace unusable sensors in most types of applications. Therefore, if a particular node is no longer running, there will be an impact on the overall performance of the network. The loss of a node may be due to various causes such as battery depletion, physical damage from environmental forces, or destruction by the enemy. If a sensor that covers a sensitive area dies and no other sensor can cover that area, the sensor array fails in its mission of efficiently distributing the sensors.

The process of deploying sensor nodes greatly influences the performance of a WSN. The problem of deployment or positioning nodes in a WSN is a strategy that defines the topology of the network, and therefore the number and position of nodes. The quality of monitoring, connectivity, and power consumption are also directly affected by the network topology.

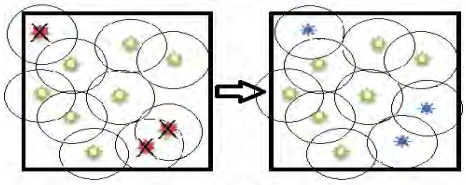
The different deployment tasks can be grouped into three main phases: A pre-deployment and deployment phase (Figure 1.3 (a)) which is achieved by the manual placement of nodes or the spreading of nodes from a helicopter, for example. A phase of post-deployment (figure 1.3 (b)) is necessary if the topology of the network has evolved, for example following a displacement of nodes, or a change of the radio propagation conditions. The third phase concerns the redeployment (Figure 1.3 (c)) which is based on adding new nodes to the network in order to replace some failed nodes. The system can iterate on phases 2 and 3.



(a) Pre-deployment and deployment phase



(b) Post-deployment phase



(c) Redeployment phase

Figure 1.3 Phases of deployment

Different issues are encountered at the level of the deployment of sensor nodes in WSN. These studies mainly concern stationary and mobile cases; single and multi-objective cases; deterministic and stochastic aspects; and finally static and dynamic deployments.

The coverage is affected by the sensitivity of the sensors represented by the detection range (noted Rs), while the connectivity is guaranteed by the communication range (noted R_c). According to (Frye et al., 2006), the degradation of the coverage probability in some WSN applications is tolerable, whereas the degradation of the probability of connectivity could be fatal for the network. In the context of dynamic deployment, we investigate in (Mnasri et al., 2014a) the research works proposing solutions for the sensor nodes repositioning and its related issues. Connectivity and coverage are the most used factors in determining the communication and the detection efficiency, respectively. Moreover, we propose in (Mnasri et al., 2014a) and (Mnasri et al., 2014b) a detailed study of deployment in the static case. They distinguish two deployment methodologies according to the distribution of the nodes (either random or controlled). The treated primary objectives can be as follows:

- The coverage which is one of the most preponderant problems in WSN. Several types of coverage are presented: point coverage, area coverage, barrier coverage and coverage of an event or a moving target.
- The consumed energy and ensuring the energy efficiency.
- The network connectivity.
- The lifetime of the network.
- The network traffic.
- The reliability of data.
- The cost of deployment (depending on the number of deployed nodes).
- The fault tolerance and load balancing between nodes.

1.3 The 3D deployment strategies in WSN

When designing deployment strategies, several factors must be considered such as the monitoring area, the sensor capabilities (detection range and transmission range), the area coverage, and the lifetime. The following questions should also be considered when establishing a deployment strategy: how many sensors should be deployed? How to place sensors in the monitoring area? Where to put the sink if we can choose its position? The number of sensors can be deduced from the lower limit of the monitoring area, sensor capacity and design requirements. Three types of coverage exist:

1.3.1 Full coverage

Full coverage is achieved if every point in the 3D RoI is at least covered by a sensor. Full area coverage requires the deployment of a large number of sensors, which increases cost and complexity. However, partial coverage only guarantees a certain percentage of coverage (Wang et al., 2007). Figure 1.4 illustrates different geometric layouts used in the 3D coverage.

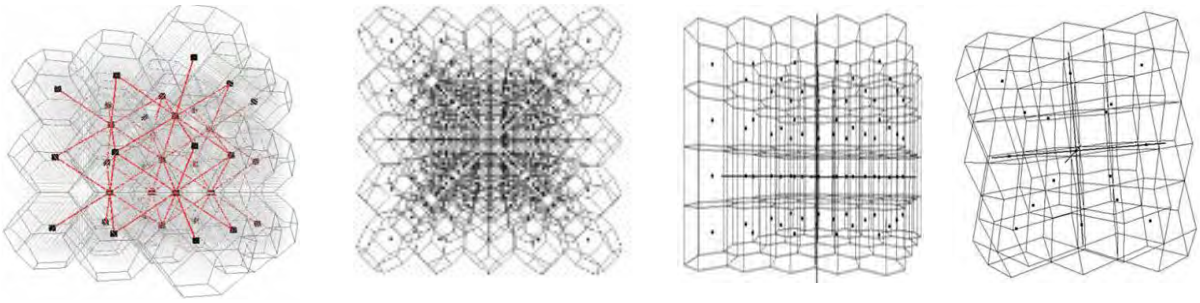


Figure 1.4 Different geometric 3D coverage layouts (Younis and akkaya, 2008)

1.3.2 K-coverage

With regard to the k-coverage, a 3D RoI is assumed to be k-covered if there are at least k sensors that cover and monitor each point of the 3D RoI. Indeed, k-coverage represents the logical extension of the case of 1-coverage. In the case of k-coverage, the distance between the detection nodes is minimized with the appearance of overlaps between the detection spheres. In general, when achieving the k-coverage optimally (K>1), the complexity of the coverage algorithm will increase (Xu et al., 2010).

1.3.3 Surface coverage

In the case of surface coverage (shown in Figure 1.5), the sensors can only be deployed on the surface of the RoI. Many real applications require this kind of surface coverage where the RoI is a complex surface that is neither a complete 3D space nor a 2D plane. Mathematically, the detection field can be modeled as several small simplified rectangles as a single-valued function z=f(x, y) with two node distribution models: a planar surface Poisson point model and a space surface Poisson point one (Ammari, 2014). (Jin et al., 2012) addresses the problem of surface coverage in terms of reliability of the monitored data and quality of detection.

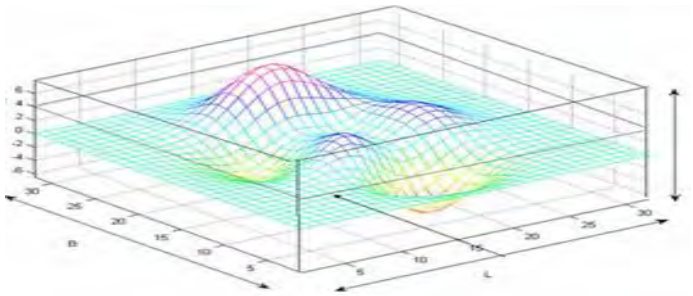


Figure 1.5 3D surface coverage (Ammari, 2014)

1.4 Conflicting types of deployments

1.4.1 Single Objective deployment vs. multi-objective deployment

The criteria and objectives, on which the deployment is optimized, are often contradictory. This is the case of coverage and energy consumption because the best the coverage is, the most the energy consumption is. It is also the case of the survivability and fault tolerance. Therefore, we cannot find a deployment that optimizes all objectives simultaneously. Hence the need for an optimal trade-off between the different objectives. According to the used approach, we can either optimize each objective alone or use an aggregation function that combines all objectives in a single function with weights that represent the importance of each objective.

1.4.2 Deterministic deployment vs. stochastic deployment

When the selection of sensor positions is possible and determined beforehand, this is referred to as the deterministic deployment. When sensors can be dropped from an aircraft, for example, this is referred to as the non-deterministic or stochastic deployment. This last type of deployment cannot be optimal as long as it can result in very dense areas and others less dense or even disconnected. Deterministic deployment provides an optimal network configuration. However, because of the size and density required to provide adequate network coverage across large geographic areas, careful node positioning is not practical. In addition, many WSN applications should operate in hostile environments which make the deterministic deployment impossible in some cases. As a result, stochastic deployment becomes the only possible alternative. As an example, the work of (Clouqueur et al., 2002) aims to determine the number of needed randomly deployed nodes for target detection by optimizing the deployment cost and the area coverage.

1.4.3 Stationnary deployment vs. mobile deployment

Considering the mobility of nodes as a criterion, two deployment strategies can be distinguished: static deployment in which nodes have unchanged positions and mobile deployment where the nodes have a mobile capacity and can move and reposition themselves after the initial deployment. Then, some regions of the network may become uncovered due to occurred events in the RoI or depletion of sensor batteries. A solution to remedy this is to move some mobile nodes to these regions. Various examples of mobile node applications can be cited such as target tracking, rescue or underwater and military surveillance. (Romer and Mattern, 2004) classify nodes according to their mobility into three classes: static nodes, mobile ones and mobile sinks. Figure 1.6 describes other characteristics of mobility: passive nodes are often attached to or carried by moving entities while active nodes have automotive capabilities. Besides, the degree of mobility varies from a continuous movement to an occasional movement with long periods of immobility.

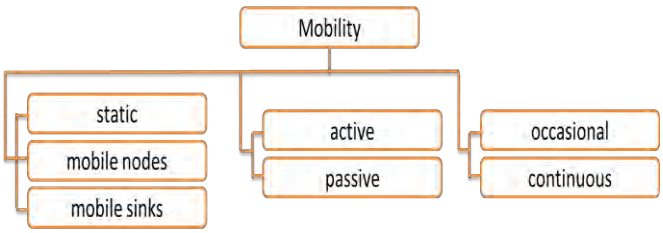


Figure 1.6 Mobility taxonomy

However, differents studies (Idoudi et al. 2012), Idoudi (2017) highlights the necessity to optimize the movement of nodes. According to (Guvensan and Yavuz, 2011), moving a node one meter can consume more than 30 times of energy than transmitting 1 ko of data. Despite this, mobility provides better coverage and connectivity and increases the network adaptability.

1.4.4 Homogenous deployment vs. heterogeneous deployment

The type of application for which the WSN is installed determines the types of sensors to deploy: are they homogeneous or having different roles thus characterizing a heterogeneous architecture. A wireless network is qualified as heterogeneous if the distribution of the nodes in the RoI is heterogeneous. Indeed, the first visions of the WSN consider homogeneous devices, generally identical from software and hardware point of view. However, different recent applications require a variety of devices that differ in number, type and role played in

the network (some nodes may function as cluster heads or gateways to other communication networks such as GSM networks, satellite ones, LoRa, SigFox, 5G, NB-IoT or WiFi).

1.4.5 Static deployment vs. dynamic deployment

Static deployment assumes that there is no mobility after the first positioning of nodes in the network. In the case of static deployment, the best node locations are chosen based on the optimization strategy, then no changes will occur during the lifetime of the network. Static deployment can be either random or deterministic. As we mentioned before, the positions of nodes have a considerable influence on the performance of the WSN and its operation. On the other hand, metrics which are independent of the state of the network and assuming a fixed operation scheme are often used to choose the locations of nodes. Among these static metrics, the distance between nodes. Moreover, the initial configuration of the network does not take into account the dynamic changes during the operation of the latter such as the mobility of targets. Hence the interest of dynamically repositioning nodes during the network operation. For example, to substitute some nodes having exhausted batteries, other sensors can be moved. Dynamic deployment is a solution for the problem of no guarantee of optimality during the initial deployment. It assumes that nodes can move in a coordinated way in the RoI. However, the relocation of nodes during the network operation is often expensive and requires the continuous monitoring of the state of the network and the events occurring in the neighborhood of the node. In addition, solutions must be found to the problem of data manipulation and delivery during the relocation process.

1.5 Initial objectives to be satisfied during deployment

1.5.1 The number of deployed nodes, cost and scalability

- Number of nodes: More than half of the papers dedicated to resolve the optimal deployment of nodes aim to reach the defined goals with a minimum cost. These studies are based on the assumption that low-cost detection devices embedded inside the sensors are used to monitor phenomena such as temperature, pressure, humidity, light, sound or magnetism. However, if we consider the deployment of hundreds or thousands of sensors, the overall cost of deploying the network must be taken into consideration. Thus, the number of sensors deployed is one of the essential measures that must be taken into account in the WSN deployment process. Indeed, in some specific applications, it is not realistic to consider low cost sensors because the more sensors used, the higher the cost becomes. Hence, the cost of a sensor strongly depends on the target application and the monitoring environment. For example, in the case of oceanographic applications such as offshore exploration, the cost of the sensor will be much higher because the detection device is much more sophisticated in order to detect specific phenomena and withstand the influences of the environment.

- Cost: The cost of a WSN starts from the construction phase of nodes. This cost may be variable depending on the application and the included detection devices. Nodes can be equipped with other equipment such as GPS that can increase the final cost of the node. Some applications do not take into account the cost of the node because of the low price of the used nodes. In this case, random deployment is an excellent way to deploy nodes in the RoI. A forest, an ocean or a battlefield are some examples of these applications. In other applications, the node costs are quite high and must be included in the node deployment strategies. Besides the node construction cost, deployment and maintenance costs are also applied to the total cost of a network. When the deployment is not random and is done by hand or with the help of particular automated robots, the cost of placing nodes must be taken into account. Indeed, the more the network needs nodes, the more its construction, deployment and maintenance will be high.

- **Scalability and density of nodes**: To monitor and detect events, some WSNs require hundreds or thousands of deployed nodes. In some specific applications (see section 1.8), this number can even reach an extreme value of millions. Scalability is another essential metric to consider when the deployment scheme is designed because it will affect the network coverage, cost and performance. High-density areas increase the cost of networking and computing. While low-density areas can cause the problem of coverage holes or network partitions. The distribution of the nodes may be uniform or non-uniform. Indeed, sensors that are near the base station are more likely to be used in data transmission. Thus, the density of nodes near the base station must be higher than other areas. In this case, the density of the nodes is said to be non-uniform. Uniform density of nodes decreases the likelihood of nodes clustering and the appearance of cover holes. The number of nodes must be minimized while ensuring maximum coverage.

1.5.2 Coverage problem

Coverage and deployment issues are fundamentally related. In order to achieve deterministic coverage, a static network must be deployed in a predefined form. Thus, optimal deployment of sensors will also provide good coverage of the monitored area. The coverage can be defined as follows: if each point in the region is at a maximum distance of R_s (detection range) from at least one sensor, then the network guarantees full coverage. Depending on the application for which the network is deployed, different levels of coverage may be implemented. Some applications (see section 1.8) accept a deterioration of coverage. While some other critical applications require full coverage over the lifetime of the network and failure to meet this requirement can lead to network failure. The ratio of the area covered by the detection nodes to the global application area is defined as the coverage. The ideal value of this parameter is 100%, which means that the entire surface is covered by nodes. In addition, the degree of coverage of a point in the RoI is defined as the number of nodes covering that point. Based on this definition, the degree of coverage of a WSN is defined as the minimum of the degrees of coverage in all the RoI. Theoretically, each node has a communication range (R_c) that defines the field in which another node can be placed to receive data. R_c is different from the detection range (R_s) that defines the area that a node can observe. R_c and R_s may be equal but are often different.

The coverage in WSN is widely discussed in the literature, especially in the 2D case. Generally, the considered coverage issues are area coverage, point/target coverage, energy optimization coverage, and k-coverage. The percentage of the deployment area in which an event can be monitored by at least one sensor is expressed as 1-coverage (Ammari M. Habib, 2014). In an ideal deployment, the coverage must be 100%. Unfortunately, sensors are subject to failure, measurement errors, damages and energy depletion. Therefore, a more general case has been defined: the k-coverage ($k \geq 1$), where each point of the monitored area must be monitored by at least k sensors.

The most studied coverage problem is that of area coverage. The primary objective of the sensor network is to cover and monitor an area (also called region). Figure 1.7 shows an example of several types of sensor deployments to cover a square-shaped area.

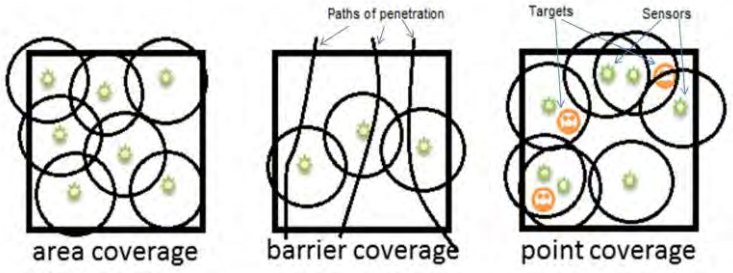


Figure 1.7 Coverage types

In the area coverage problem, each sensor covers a particular sub-area, and the total area of the sensor network is constituted by all the covered sub-areas. Maximizing the total coverage area of the WSN is the primary objective of the coverage area problem. Area coverage problem is closely related to the targeted application, such as target detection and tracking, battlefield monitoring, personal protection, and animal behavior tracking.

The most commonly used model for coverage is the disk detection model. It is based on the idea that all the points located in a centered-on-sensor disk, are supposed to be covered by the sensor. However, in the literature, many research works assume a fixed detection range and an isotropic sensor detection capability. The detection capability of a sensor can be classified as a binary or probabilistic. Some published articles, use the ratio of the covered area and the overall deployment area as a metric to measure the quality of coverage. However, other works focused on the worst-case coverage. In addition, another less used method is based on measuring the probability of movement of targets or the probability of an event that is happening without being detected.

Most coverage problems in WSN are related to the quality of area monitoring or event tracking. When the nodes are randomly deployed, the position of the nodes is no longer controlled. Therefore, some places in the target region remain uncovered. These areas are known as coverage holes. In some applications, nodes are equipped with mobile units to be able to move and cover these holes. To measure the coverage, we can just divide the RoI following a grid of small squares where each one represents a sensitive zone. Each square containing a node is considered as covered while each square that does not contain a node is considered as uncovered. Using this measurement method, the percentage of covered squares of all squares is known as the coverage rate. (Aitsaadi et al., 2011) considered this model to measure the coverage rate found by their proposed algorithms. The circular detection model can be also adopted. It is also known as a binary disk model where each point inside the cover disc centered on the node is covered, and every point that is outside that disc is uncovered. Due to the characteristics of the real world, this model was unrealistic and the researchers proposed a more realistic model based on probability. According to these models used to represent the detection capabilities of each node, the detection coverage for the entire network can be determined. The probabilistic circular model is more realistic (but more complicated) then other models.

1.5.3 Connectivity problem

In addition to the coverage, it is crucial for a WSN to maintain connectivity. Connectivity can be defined as the ability of nodes to reach the base station. If there is no available path from any node to the sink, then the data collected by that node cannot be processed. Regardless of the used communication model, network connectivity is often measured by a connectivity graph. If this graph is full connected, the network is assumed to be connected; otherwise it is considered as partitioned. That is, a network is said to be connected if active nodes can communicate together and a path can be established between each sending node and a presumed receiver one. Otherwise, we may not be informed of the events occurring in the RoI. This communication can be indirect using other nodes as relays. If a sensor detecting a fire cannot send this information to the base station, it will be more challenging to extinguish this fire. Therefore, the network topology must be connected and able to give information about any evolution of the monitored events.

Network connectivity is a problem to consider when designing the WSN. Indeed, in different research works; contrary to the coverage that is often considered a constraint or an objective when positioning nodes; the network connectivity has been neglected assuming that a good coverage will always give a connected network (when the communication field

exceeds the detection one). Despite this, a limited communication field results in a reduced connectivity unless there is a substantial redundancy of coverage.

The connectivity also makes possible the adjustment of the communication between the nodes of the network if a part of the network becomes disconnected due to breakdowns or energy exhaustion of a set of nodes. Connectivity is also needed to ensure the message propagation to the base station. The loss of connectivity often involves the end of the network lifetime. Assuming that there are several distinct paths between every two nodes, the degree of connectivity can be defined as the minimum number of paths between every two nodes in the network. If we should be connected with more than one sensor, this is referred to as the k-connectivity (Guang et al., 2009). As well as the coverage, various research works describe based-on k-connectivity algorithms that construct k distinct paths between each node and sink. K-connectivity introduces more reliability than 1-connectivity in data transmission by guaranteeing k-1 other paths if a path fails. Moreover, using different paths facilitates the design of distributed mechanisms such as the traffic load balancing which helps to reduce power consumption and extend the network life.

1.5.4 Coverage with Connectivity

Coverage and connectivity are related and the locations of nodes affect them both. Any optimal deployment strategy must simultaneously ensure a globally connected network and an optimal coverage. (Bai et al., 2006) formulate sufficient conditions to ensure coverage and connectivity in WSN.These conditions are influenced by the node locations, the detection range (R_s), and the communication range (R_c). They prove that if $R_s \geq \sqrt{3}R_c$ and the RoI is fully covered, then the communication graph is connected. Thus, it is necessary to combine the achievement of coverage and connectivity in a single deployment planning.

- **Relation between 1-coverage and connectivity**: (Zhang, 2003) and (Wang et al., 2003) were the first to integrate the scheduling of activities in detection and communication. Both studies state that if a convex region is totally covered entirely by sensors, the corresponding communication graph will be connected when $R_c \geq 2R_s$. That is, if $R_c \geq 2R_s$, the network can guarantee both coverage and connectivity.
- **Relation between k-coverage and connectivity**: Based on the condition $R_c \geq 2R_s$ for coverage to imply connectivity, (Tian and Georganas, 2004) study the relationship between the degree of coverage and the connectivity. They prove the following findings for a convex region *A* covered by K nodes if $R_c \geq 2R_s$: *i)* These nodes form a connected communication graph K. *ii)* Interior connectivity is 2K. *iii)* It is possible to disconnect a boundary node from the rest of the communication graph by removing K sensors.

1.5.5 Energy efficiency

Generally, nodes in WSN have limited energy. The initial deployment scenario is usually based on a set of nodes and a smaller number of base stations. Indeed, a mobile node has different components such as memory, battery, processor, detection devices and mobility ones. These components use the battery power. After deployment, it is not always possible to recharge and maintain the node battery. Therefore, it is beneficial to know the energy consumed by each of these components to optimize their energy consumption. In this respect, (Pei et al., 2008) investigate the energy consumption of different types of node's hardware for different applications with different microprocessor platforms and communication protocols. (Mei et al., 2004) have also proposed a mobile device for moving nodes and calculated the energy consumed by nodes when moving straight or turning.

The energy consumed by nodes is not uniform. In general, the nodes near the BS consume more power than nodes far from BS or at the border of the RoI, because they are used as relays to transmit the packets to other nodes. In many practical scenarios, the placement of sensors can be controlled so that the density of nodes can be changed by the

variation of the energy consumption. In this way, the lifetime of the network can be increased. In a small WSN, nodes can be arbitrarily placed. As for a large WSN, it is possible to deploy several sensors in areas where the energy consumption is much higher. For example, in the case of aerial deployment, by merely dropping sensors on a set of selected areas. Thus, it is necessary to establish a correlation between the density of nodes and the energy consumption. This density must correspond to the distance between the sensors and the BS.

Indeed, the waste of energy can be caused by several reasons: *i)* The phenomenon of collision: if more than one packet is received by a node at the same time, these packets must be discarded and a process of retransmissions must be initiated. *ii)* The over-listening where the node receives packets that are destined to other nodes. *iii)* The control packet overhead.
The energy consumption is often considered interchangeable with the lifetime. Indeed, due to the limited energy resources in the node batteries, it is necessary to effectively deploy and use the nodes in order to increase the lifetime of the network.

1.5.6 Network lifetime

As mentioned above, the limitation of the energy resources affects the overall operation of the WSN. Therefore, it is important to optimize the energy consumption to maximize the network lifetime. This latter can be defined as the time after which the network is partitioned in a way that makes data collection impossible from a part of the network (Ferentinos et al., 2007). Another definition of the network lifetime uses the time after which the first node becomes non-operational. Indeed, according to the application of the WSN, the lifetime varies from a few hours to several years. It considerably influences the energy efficiency and the robustness of the nodes. In order to increase the lifetime of the network, different ways exist such as the relocation of nodes, the incremental deployment (adding new nodes after identifying defective ones), the balancing of loads and energy consumption between nodes to minimize the risk of node failure.

1.5.7 Network traffic

Another factor that directly affects the lifetime is the message traffic. Indeed, the consumed energy is proportionally dependent on the number of delivered messages. Hence, the message traffic must be minimized in WSN. Two types of messages exist. *i) Network messages* that contain information such as viewing angle, status, residual energy, detection range, and node position. *ii) Application-related messages* that contain data detected from the environment. To optimize the lifetime in WSN, the distribution of redundant messages must be avoided or minimized. To achieve this, we often use the in-network processing in the case of *application messages* (Ammari M. Habib, 2014). Regarding *network messages*, they are exchanged during the initial configuration where each node determines its position and those of its neighbors by using specific network messages. Repositioning strategies aim at calculating the final position of nodes. These strategies are iterative, which implies excessive message traffic. Two repositioning approaches exist: repositioning with physical movement and with virtual movement. Inversely with the virtual movement strategy, in the physical movement strategy, the sensor nodes physically change its position after each step. In terms of network lifetime, the physical movement strategy is less efficient than the virtual one.

1.5.8 Data fidelity

Data fidelity is another important factor to be considered when deploying and designing WSN. This factor aims to ensure the credibility of the collected data. WSN determine an evaluation of the detected phenomena by collecting readings from different independent nodes often heterogeneous. The merging of these data decreases the likelihood of false alarms and increases the reliability of the reported incidents. From the point of view of signal processing, the merging of these data aims to minimize the effect of distortion by including

the ratios of different nodes in the evaluation of the detected phenomena. Redundancy of nodes in the RoI increases the accuracy of the merged data, but it has the disadvantage of requiring an increased density of nodes which causes additional costs.

1.5.9 Fault tolerance and load balancing

As noted above, excessive power consumption causes the malfunction of nodes that lose their energies and becomes useless. Hence, the WSN must be fault tolerant. In this regard, fault-tolerant deployment strategies are used to prevent individual failures that minimize the overall network lifetime. Different deployment strategies can be used to make the network more fault-tolerant. Among others, the distribution of loads between nodes, the deployment of new nodes and the relocation of them. More details on the problem of fault tolerance are discussed in (Wang et al (2005).

1.5.10 Latency

Latency mainly concerns the delay in the aggregation, transmission or routing data. It is measured by the time elapsed between the departure of data packets from the source node and the arrival of these packets to the destination. Indeed, one of the causes of the emergence of latency is the use of alternative paths which are often longer than the main path. This leads to a higher consumption of energy. Moreover, some applications merge data to reduce the network traffic, which causes latency in the network.

1.6 Sensing models

In this section, we present the most important used detection models, their differences and their cases of use.

1.6.1 Binary model

In the binary model, the detection field of each node is considered as a circular area and a target can be either detected or undetected. In this model, there is no transition period and a slight displacement of the target may lead it to be outside the detection field. This model aims to simplify the analysis of detection. Unfortunately, it does not realistically reflect the detection capabilities of nodes. It only takes into consideration the distance between the sensor and the target. This model is illustrated in Figure 1.8.

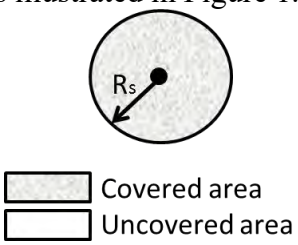


Figure 1.8 Binary sensing model

Mathematically, if the node S is at the location (x_s, y_s), the detection field of S is a circular area of radius R_s centered in (x_s, y_s). In the binary model, S detects the target in its detection field with a probability of 1 because the distance between the target area and the node is less than R_s. It cannot detect this target outside its range detection (a probability of 0). Hence, the probability P_{sb} of detecting a target T at the position (x_t, y_t) by S in the binary model is as follows:

$$P_{sb} = \begin{cases} 1 & D_{TS} \le R_s \\ 0 & Otherwise \end{cases} \quad (1)$$

The distance between T and S, denoted D_{TS}, is calculated as follows:

$$D_{TS} = \sqrt{(x_s - x_t)^2 + (y_s - y_t)^2} \quad (2)$$

1.6.2 Probability model

In the probabilistic model, there is a transition period between the ability of a node to detect a target and its non-ability. Hence, the target is detectable with a probability that varies between 0 and 1. By deploying a WSN, the distance that separates the target to be monitored from the node and the characteristics of the node itself influence the perception of the node and its ability to detect the target. Indeed, the probability of detection of an event is inversely proportional to the distance separating the node from this event. However, in this model, the geometric distance is not the only factor used to determine the coverage. Other factors come into play such as noise, environment, interference, and signal attenuation (Wu et al., 2007). The probabilistic detection model, shown in Figure 1.9, reflects a concept of fuzzy coverage that represents a changing regularity of sensor detection capability.

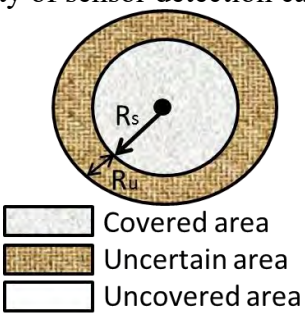


Figure 1.9 Probability sensing model

According to figure 1.9, we assume the existence of two critical distances for each node. A detection distance R_s (same parameter as the binary model) where the target is detected with a probability of 1 if the distance between the target and the node is less than R_s. The second distance is R_u which represents an uncertain detection range. The probability of detection of the target by the node depends on the distance that separates them if the distance between them is between R_s and R_s+R_u. The target is not detectable by the node if the distance between the target and the node exceeds R_s+R_u. Mathematically, this probability can be defined as follows:

$$P_{sb} = \begin{cases} 1 & D_{TS} \le R_s \\ e^{-\lambda \alpha^{\beta}} & R_s < D_{TS} \le R_s + R_u \\ 0 & R_s + R_u < D_{TS} \end{cases} \quad (3)$$

Knowing that $a = D_{TS}-R_s$. λ and β are constants that depend on the sensors' hardware properties.

1.6.3 Tracking detection model

As an extension of the probabilistic model of event detection described in the previous section, Aitsaadi et al. (2011) proposed a new model that highlights the duration of the event. They assume that the probability of detection of an event is increased when this event occurs for a long time at the same point. This model is ideal in the case of a target tracking application. They propose a variable *t* that quantifies the duration of an event at a specific point. Let *T* be a period at the end of which the detection algorithm is executed and a decision is made if an event occurs. Hence, the probability of detection of this event will be calculated as follows:

$$P_p^s(t) = \begin{cases} P_p^s & if\ 0 < t \le T \\ 1 - \left[1 - P_p^s\right]^{\lfloor t/T \rfloor} & if\ t > T \end{cases} \quad (4)$$

1.6.4 Coverage and connectivity model

Coverage diagrams describe the topology of the sensor network to ensure perfect and optimal detection while connectivity schemes are related to the transmission and reception of messages in the network. According to (VAL, 1994), coverage and connectivity schemes include:

- "Connectivity-first" or "Guaranteed-connectivity" scheme (figure 1.10 (a)): it aims to guarantee connectivity. Thus, sensors are at a distance equal to r_c. According to (Wang et al, 2008), the efficiency of this deployment mode is guaranteed when $R_c \le \sqrt{3R_s}$, R_c is the communication range and R_s is the detection one.
- "Coverage-first" or "Guaranteed-Coverage" scheme (Figure 1.10 (b)): This deployment scheme tries to reduce the number of sensors by minimizing the intersections and overlaps of the coverage. In this scheme, nodes are at a distance of $\sqrt{3R_s}$. According to (Wang et al., 2008), the efficiency of this deployment mode is guaranteed when $R_c \ge \sqrt{3R_s}$.
- Hybrid coverage scheme: it is a combination of the two previous schemes. Coverage is guaranteed to a certain distance until it becomes probabilistic (Wu et al, 2007).

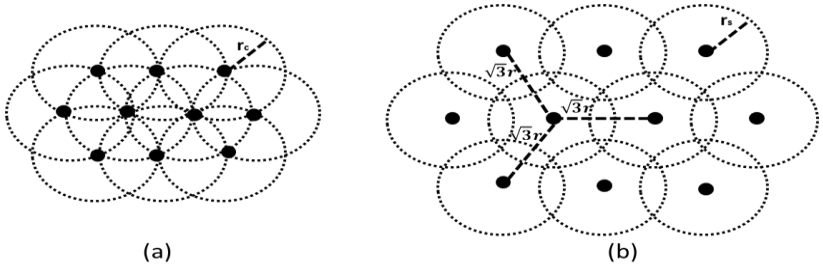


Figure 1.10 Guaranteed-connectivity scheme (a) and guaranteed-coverage scheme (b) (VAL, 1994)

Other models exist such as the model that aims to ensure better energy consumption during detection or that which aims to extend the lifetime of the network.

1.7 Open deployment issues

Although many studies have been provided in the field of optimization of nodes positioning, various research challenges remain to be solved. In what follows, we identify open research problems in the 3D sensor deployment.

- **Underwater WSN**: is a network of sensors deployed under water. Underwater WSN are investigated by different studies BOUZOUALEGH et al. (2003), BOUZOUALEGH et al. (2005). Given the cost of installing these nodes, we must minimize their number. In order to communicate under water, these networks use acoustic waves that meet different issues such as fading signal, high latency, long propagation delay and limited bandwidth. The submarine nodes must be adapted to the extreme conditions of the environment and be self-configured. Batteries installed in these nodes must be energy efficient as they are non-replaceable and non-rechargeable. Different underwater applications can be considered, including earthquake and pollution monitoring, underwater exploration and disaster prevention. Indeed, deployment problems in underwater environments are quite different from those of the WSN. In this regard, the work of (Zhang et al, 2004) provides an overview of the latest developments in

submarine deployment algorithms. The authors categorize deployment algorithms into three categories, based on the mobility of sensor nodes: static deployment, self-adjustment deployment and assisted motion deployment. Future research problems with submarine deployment algorithms may focus on the following ideas:
- How to design a mobility model that solves the problem of displacement of underwater sensors with water flow? This can make submarine deployment algorithms more efficient.
- Current deployment algorithms often deal with small scale networks while many monitored underwater areas are large scale. How to design failed node recovery mechanisms and deployment algorithms with a minimal number of redundant nodes that can be used to avoid network partitioning.
- To our knowledge, in order to extend the overall lifetime of a UASN network, little research has focused on the deployment of cyclical nodes dynamically working.
- Given the dynamism and complexity of underwater environments and the high cost of deploying submarine nodes, it is interesting to design efficient deployment algorithms for heterogeneous UASNs taking into account nodes with different ranges of communication and detection.

Figure 1.11 illustrates a deployment of an underwater WSN.

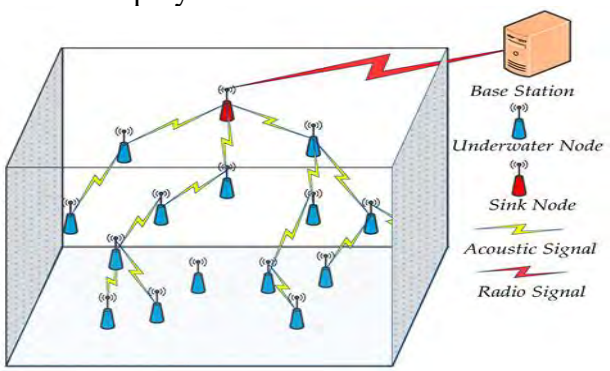


Figure 1.11 deployment of an underwater WSN (Jiang et al, 2016)

- Underground WSN: is a network of sensors deployed in mines or caves to monitor underground events. To communicate data collected from underground nodes to the base station, additional sink nodes can be deployed above the ground. Because of the attenuation and signal loss, underground wireless networks are often more expensive than terrestrial ones because of the specific equipment used to ensure reliable communication across rocks and ground. Even more, as well as underwater networks, buried node batteries are often difficult to replace and recharge which require the design of deployment algorithms and communication protocols that are energy efficient. Different applications of underground networks can be cited such as military border surveillance, agriculture and minerals.

- Multimedia WSN: is a network of sensors deployed at low cost and equipped with microphones and cameras used to detect, store and process multimedia data such as audio, images or videos. When deploying such a network, different issues need to be considered, such as high power consumption, the need for different compression and data processing techniques and the demand for high bandwidth. Moreover, it is essential to guarantee a minimum quality of service to ensure the supply of reliable content. In general, multimedia networks are used to enhance existing WSN applications, such as monitoring and tracking.

- Coordinated multi-node relocation: Different contexts require that nodes coordinate between them to efficiently manage the application requirements while having the ability to synchronize inter-relocation (Younis and Akkaya, 2008) which is the case of the exploration of inaccessible fields or the detection of landmines.

- Heterogeneity: The majority of WSN studies assume that nodes are homogeneous. In this respect, and knowing the simplicity of collection networks in IoT, it is necessary to design and further study networks with heterogeneous nodes that can be simultaneously shared by several applications with conflicting objectives.

- Simulation vs. real experimental deployment: often, studies on node deployment are validated by performance analyzes that are based on simulations. The contribution of the simulation is that it is simpler and more controllable. However, it does not reflect the modeling of real application scenarios and may be less accurate than tests based on real prototypes. It is therefore attractive to design and implement real prototypes for the analysis of network performances under realistic experimental conditions and the real validation of the proposed models for deployment and coverage. Our contributions in this thesis are validated by real experiments. The results of these real experiments are then compared to simulations to draw more consistent and applicable findings in different contexts.

- **Mechanisms for detection and repair of coverage holes**: Despite the fact that different works have addressed the problems of connectivity and coverage, the issue of coverage holes has not been well highlighted. In this context, the problem of the coverage holes in the detection field is related to the deployment. Indeed, coverage holes are usually caused by failures of sensor nodes, by hostile environments (battle areas or volcanic regions) or by random deployment of stationary nodes in hybrid WSN composed of static and mobile nodes. Mobile sensor nodes are often added after the initial deployment to overcome the problem of coverage holes. However, because of the low power of mobile nodes, an effective management of their movements to maintain the network coverage and connectivity while minimizing the power consumption becomes a challenge. In this context, (Guvensan and Yavuz, 2011) solved the coverage problem in directional WSN. Indeed, directional nodes are often equipped with ultrasonic sensors, video sensors or infrared sensors. They differ from traditional omnidirectional nodes in different parameters such as viewing angle, direction of operation and field of view. (Guvensan and Yavuz, 2011) classify existing approaches solving the problem of coverage holes and determine their complexities, specificities and performances. They classify coverage optimization methods into four main classes: targeting-based, coverage-based, coverage guaranteeing connectivity-based and extending lifetime-based. They define the detection models, the challenges envisaged for DNS (Directional Sensor Networks) and their (dis)similarities to WSN. (Shen et al., 2013) develop an adaptive algorithm named AHCH (Adaptive Hole Connected Healing), to solve holes problems with the guarantee of network connectivity without the need to find new deployment schemes. Indeed, this algorithm adapts the existing deployment scheme to avoid coverage holes. To prove the effectiveness of this algorithm, authors compare, for different time intervals, the optimal solution with the estimation of the adaptive approximation ratio of this algorithm, with a complexity in O(log|M|), M is the number of mobile sensors. Then, they extend this algorithm in the general case by establishing two other versions to solve the same problem. The first version is InAHCH (Insufficient AHCH) which is used to solve the problem of holes if the number of mobile sensors is insufficient to guarantee the k-coverage for all the holes. The second version is GenAHCH (General AHCH) which is a generalization of the specific cases treated by the AHCH algorithm.

- **Three-dimensional node positioning**: The two-dimensional coverage problem has been solved in (Zhang et al., 2010) using an algorithm with a polynomial time in terms of the number of sensors. On the other hand, concerning the tridimensional problematic, it is much

more complicated to solve; and the use of random deployment is often inefficient in the case of real 3D networks. Moreover, compared to 2D coverage, the complexity of 3D coverage algorithms exponentially increases compared to the number of sensors (Zhang et al., 2010). So far, most studies on node positioning strategies are limited to the two-dimensional networks. Similarly for networks with multimedia applications, the focus has been on coverage by managing the angular orientation of the nodes in a 2D plan. Indeed, most deployment and coverage algorithms used in 2D spaces become NP-Hard in a three-dimensional space (Akyildiz et al., 2007). Yet, with the emergence of WSN applications in underwater and aerial surveillance, solving issues related to connectivity and 3D coverage has become a necessity. Hence, sensor deployment optimization strategies in large-scale 3D WSN applications is one of the most emerging research topics currently. In this thesis, we propose to solve this problem using evolutionary optimization algorithms. In this regard, different related works are detailed in section 1.11.

1.8 Applications of the deployment

Technological advances on the construction and manufacture of sensors have made them smaller, more affordable and more reliable which allowed broadening the range of targeted application areas. The integration of WSN technology in industry has improved the business performance. Still more experimental work is needed to increase the reliability and efficiency of real-world WSN use. Given the variety of sensors, the WSN fields of application are very varied. Generally, WSN applications concern the detection of an event or a phenomenon, taking periodic measurements (physical quantities), the detection of contours or the tracking of a moving target.

- **Ecological Applications**: Another use of WSN is optimizing the use and consumption of energy resources, such as a sensor incorporated into an air conditioning system in a building. Indeed, this network manages the air-conditioning according to the location of the individuals. The air conditioning trips only if it is necessary (when the sensor detects the existence of people in the RoI). This is the case for a heating, lighting or ventilation systems.

- **Traceability and localization applications**: WSN can be a major fault correction of Global Positioning System (GPS) systems which are high energy consumption. Indeed, people who ski can be equipped with sensors. These sensors will be handy to locate victims under snow in case of an avalanche. Several other traceability and localization applications with significant economic interests can be considered. Again, unlike location solutions that are based on GPS systems, WSN allow their deployment in enclosed areas such as mines or underwater areas (Baouche et al, 2009). Further inverstigations can be found in Fofana et al. (2016) and Van den Bossche et al. (2016).

- **Industrial applications**: The use of WSN in recent contexts is very beneficial since the use of a wireless network will significantly reduce the deployment and maintenance constraints. WSN can enhance manufacturing activities and it is ideal for any activity that requires fixed location and limited resources. Several other constraints and restrictions can be solved using WSN. Among others, weight restrictions (in an airplane for example) and mobility ones (tracking a moving target or detecting a robot movement). Among industrial WSN applications, we can mention the following ones:
- Surveillance and monitoring activities on hazardous conditions (radioactivity exposure, etc.).
- Monitoring polluted geographical areas or real-time monitoring of a contaminated area to draw a dynamic geographical map. As an example, we cite the OCARI (Open Communication protocol for Ad hoc Reliable industrial Instrumentation) project which is a

low-power wireless communication protocol based on the IEEE 802.15.4 standard. It was initially developed by the ANR OCARI project funded by the French National Research Agency.
- Monitoring the operation of machines.
- Logistics and inventory control.
- Process monitoring and Traffic monitoring.
- Preventive maintenance of equipment and structures.
- Supervision of foundations and buildings in civil engineering

- **Security applications**: WSNs have a beneficial use in the security field. Indeed, they can limit the financial expenses of securing structures. Among the WSN security applications:
    - The deployment of sensors for detecting cracks and alterations in structures and buildings, following an earthquake or just to control its aging. This can anticipate the destruction of the structure.
    - Monitoring of movements in a geographical area to set up a distributed system of intrusion detection. The distributed aspect of different systems makes it possible to cross it or put it in a state of dysfunction.

- Environmental Applications: To ensure broad coverage over a large geographic area, WSN can be used. We can cite several typical WSN environmental applications:
- Dissemination of temperature-sensitive sensors on a forest (deployment from an airplane) to detect the outbreak of fire.
- Controlling and managing the irrigation of green surfaces by detecting dry areas.
- The deployment of sensors on industrial sites, nuclear structures or oil refineries to capture and report the existence of leaks of toxic products (gases, chemicals, etc.) for rapid and effective intervention.
- The control of natural parks: sensors can be deployed to monitor animal movements and activities or to report and provide information on seasonal migrations of birds.

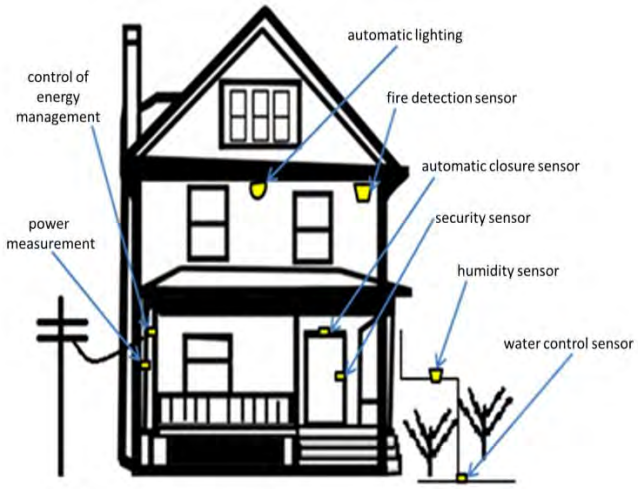
- **Medical and veterinary applications**: In medicine, WSN are generally used to monitor patients permanently. They are deployed to collect physiological data on vital functions such as heart rate, glucose level, respiration or blood pressure which facilitates the diagnosis of diseases and the monitoring of patients' health status. These sensors can be implanted under the skin or worn by the patient. As applications for this field we can mention:
- Micro-cameras are providing images of the interior of the body without performing a surgery operation.
- Biomedical micro-sensors forming an artificial retina to correct vision.
- Micro-sensor for the study of phonation.
- Early detector of cutaneous infection in peritoneal dialysis.
- Non-invasive micro-sensor for the study of the autonomic nervous system.
- Micro-sensors for the early detection of certain diseases such as cancer.

- **Military applications**: They were the first engine for the development of WSN. These applications take advantage of specific features of WSN such as easy deployment, fault tolerance and auto configuration. In military field, these networks are often deployed to monitor or analyze strategic areas, unattainable ones or those of the enemy to predict its activities and movements. Several other contexts of use can be found such as: Providing information regarding the loss or damage after a battle, monitoring the enemy equipment and ammunition, detecting chemical or nuclear attacks, etc.

For military applications, submarine deployment is one of the most interesting applications. Due to the complexity of the deployment environment in 3D spaces and the specific characteristics of underwater acoustic channel, many factors must be taken into consideration.

- **Commercial applications**: WSN can also be used in a commercial context. Indeed, sensors can control the operations of storage and delivery and provide (in real time) information about conditions, directions or positions of the goods. In addition, we can cite the WSN used to follow the production chains in factories and to control the entire production process.

- **Smart homes :** Another very practical WSN use is that of smart homes (Figure 1.12 (a)). Figure 1.12 (b) illustrates the intelligent home of the IUT of Blagnac in Toulouse in which our experiments are carried out. This application mostly targets people with disabilities or elderly to ensure their safety. Among the services offered in this context, the automatic control of doors and curtains (closing and opening), the control of household appliances (activation or shutdown), the triggering of the watering of plants in the garden or the management of air-conditioning and heating systems.



(a) Illustration of a smart home

(b) Our smart home (Blagnac, Toulouse)

Figure 1.12 Smart homes

Table 1 illustrates different recent deployment applications.

Table 1.1 Recent deployment applications

Paper	Application(s)
Qu (2013)	Mobile WSN, indoor surveillance applications
BOUSSAÏD (2013)	Signal detection Segmentation of grayscale images by multi-level thresholding
Aitsaadi et al, (2011)	Image processing ; 3D modeling
Guvensan et Yavuz (2011)	Directional WSN
(Han et al, 2013)	Acoustics WSN; submarines WSN
Ben Hadj et al. (2006)	Surveillance of the elderly persons and smart homes
Brinis and Saidane (2013)	Precision agriculture

1.9 Methodologies for solving the problem of deployment

1.9.1 Virtual Force algorithms

VFA (virtual force algorithms) are used to solve the problem of coverage and deployment of nodes. Different studies aim to resolve the deployment problem using VFA such as (Boufares et al., 2015).VFA consider nodes as a set of points that exert forces of attraction and repulsion between them. This calculated force permet to produce an acceleration allowing the nodes to

move. The authors in (Yu el al., 2013) solve the deployment problem in mobile sensor networks using the *Van Der Waals* force. It relies on a friction force that is introduced into the force equation using an adjacency based-on Delaunay-triangulation relationship between nodes. To evaluate the uniformity of the node distribution, VFA use an evaluation metric called the pairwise correlation function.

1.9.2 BDA, PFDA and DDA Algorithms

The study in (Ait Saadi 2010) aims to resolve the static deployment in WSN by optimizing the quality of monitoring, the cost of deployment in terms of the number of nodes, the lifetime and the connectivity of the network. They propose three approaches of approximate resolution. The first approach is based on a Differentiated Deployment Algorithm (DDA). The second approach is based on a Bernoulli deployment algorithm (BDA) that uses a Bernoulli distribution which decides, in a probabilistic way, the addition or removal of a node in the RoI. The third approach is based on a Potential Field Deployment Algorithm (PFDA) which is a deterministic method based on virtual forces.

1.9.3 Heuristic optimization

A heuristic is an approximate method for solving an optimization problem. It is an algorithm that aims to find a feasible solution while respecting a set of constraints and criteria without guaranteeing optimality, but in a reasonable resolution time. In general, the heuristic methods consist in exploring a search space to maximize or minimize a given objective function. Different heuristic resolution methods are used depending on the complexity (in size and structure) of the targeted search space. A small search space requires a deterministic method, whereas a larger and complex search space requires an approximate research method (genetic algorithm, simulated annealing ...). The heuristic solution is considered by a set of variables (decision variables describing a system for example). These variables are defined on continuous or discrete domains. If it is a discrete domain, then we speak of combinatorial optimization and the set of solutions is enumerable. If some variables are related to each other, we talk about continuous optimization. Scheduling problems are part of constrained combinatorial optimization problems. In terms of complexity, the majority of scheduling problems are NP-hard. Indeed, one of the solutions to solve them is to develop heuristics that provide feasible scheduling in a reasonable time with a duration close to that of an optimal scheduling. The study in (Zhang et al., 2014) proposes a deployment based on swarm intelligence algorithms such as the particle swarm optimization algorithm, the bee algorithm and the ant colony algorithm. They discuss the issues of self-organized networks at the physical, MAC and network layers.

1.10 Other problems similar to the node positioning problem

Having similar constraints and issues, several problems can be compared to the 3D indoor deployment. Among others, the following problems can be envisaged:

1.10.1 Set MultiCover Problem

The Set MultiCover Problem (SMCP) is a generalization of the Set Cover Problem in which each element *i* must be covered by a minimum number of sets *ri*. This constatation involves covering all elements while minimizing costs (Fujito and Kurahashi, 2006). The SMCP problem can be reduced to a PP (Positioning Problem) as follows: Given an instance of the SMCP problem, we can build an instance of the PP such that an optimal solution of the PP is also an optimal solution of the SMCP.

1.10.2 Art Gallery Problem

There is also a great resemblance between the problem of sensor positioning and the problem of the Art Gallery Problem (AGP). The AGP has been defined and treated by (O'Rourke et al, 1987) which involves placing agents or surveillance cameras in an art gallery represented by a non-convex polygon of *n* vertices in such a way that each point of the gallery is visible to at least one agent or camera. The AGP problem has been solved for a 2D surface and has been demonstrated to be NP-hard for a 3D surface. Different variants of the AGP problem have been studied. An interesting version of the AGP problem is that after a random initial deployment, how agents move so that each point is visible to at least one agent, this is the Distributed Art Gallery Deployment Problem. However, in some positioning problems, the sensors can be heterogeneous unlike the AGP problem where cameras have the same characteristics. Cardei and Wu (2006) study the problem of energy optimization for the coverage problem in WSN. They detail the WSN design factors and present coverage issues similar to those of coverage in WSN. They are particularly interested in AGP, *Ocean Coverage* and *Robotic Systems Coverage*. The addressed energy optimization issues are either based on energy efficient area coverage or on energy efficient point coverage.

1.10.3 Other similar problems

A set of academic problems has a close relationship with the problem of sensor deployment. For example, the problem of warehouse location which consists of setting up a set of warehouses to serve a certain number of demand points while minimizing costs. The k-center problem is one of the problems of warehouse localization and more specifically of the minimax location-allocation type (Hale and Moberg, 2003). Many works have concentrated on studying different versions and their complexity. The k-center problem defines an objective function of minimizing the maximum distance between a demand point and the nearest warehouse. The decision version of this problem is that of determining whether a set of request points can be covered by k centers at a distance equal to r. Another similar problem is that of k-median. Formulated by Hakimi in 1964, the k-median problem is one of the minimum localization-allocation problems (Hale and Moberg, 2003). It is a question of determining k centers among a set of n centers such that the sum of the distances between each point of the request and the nearest center is minimum.

1.11 Comparaison and review of recent studies on WSN deployment

In literature, different works aim at resolving the problem of deploying sensors in WSN exist. Table 2 illustrates a comparison between the recent optimization approaches used to resolve the deployment problem in WSN.

Table 1.2 Comparisons between recent works resolving the deployment problem in WSN

Paper	Application	Space	Sensing model	Contribution / Approach	Objective(s)	Disadvantage(s)
Konstantinidis et al. (2010)	Power assignment and deployment problem	2D	Deterministic	A modified MOEA/D algorithm	Maximize the coverage and the lifetime	The performance of the MOEA/D is compared with only one algorithm (the standard NSGA-II) which does not give a sufficient idea about the effectiveness of the proposed approach compared with other ones.
Banimelhem et al. (2013)	The problem of coverage holes	2D	Deterministic	Genetic Algorithm	Minimum number and best locations of mobile nodes	No mathematical modeling is given
Aval et al. (2012)	A survey	2D	Deterministic	Multi-objective methodologies: genetic algorithms and particle swarms	Coverage, connectivity, cost and lifetime	No simulations (or experimental studies) are given to compare the proposed approaches
Qu (2013)	Redeployment	2D	Deterministic	Centralized optimization algorithms: GA and PSO	Maximize the detection range and minimize the energy consumption.	The standard versions of the proposed algorithms are tested, with enhancements only on the problem assumptions. Moreover, it would be

						more interesting to consider the 3D deployment case
Kang and Chen (2009)	3D Differentiated WSN deployment problem	3D	Deterministic	A force-driven strategy based on a multi-objective genetic algorithm	Coverage, energy conservation, detection levels	A local optimum can be given by the proposed algorithm as a final solution because of the simple proposed mutation operator
Topcuoglu et al. (2011)	Deploy multiple sensors on synthetically generated 3D terrains	3D	Deterministic	Hybrid evolutionary algorithm (three crossover operators+ a local-search phase)	Maximize visibility Maximize stealth Minimize cost	Authors do not treat in this work, the 3D WSN deployment issues with unusual detection levels that require a number of nodes that exceeds the usual detection level
Unaldi et al. (2012)	3D terrains	3D	Probabilistic	Random walk mutation + guided wavelet transform (WT)	Maximize the quality of the coverage, minimize the number of sensors	The proposed method is not evaluated in dynamic environments and using mobile sensors. Moreover, it is not proved by empirical real-world scenarios
Danping et al. (2013)	Signal and radio propagation modeling indoor and outdoor 3D scenarios	3D	Deterministic	3D multi-objective evolutionary algorithm+ a low cost heuristic	link quality, coverage, lifetime and cost of hardware	The scalability of the used algorithm is not proved
Matsuo et al. (2013)	Radio propagation model in the disaster area	3D	Deterministic	Genetic algorithm + Local search	Maximize k-coverage, Minimize the number of deployed nodes	The proposed algorithm is not evaluated by the known metrics (Hypervolume, Inverted Generational Distance).
Xu et al. (2014)	Deploy indoor wireless antenna with obstacles	3D	Deterministic	Mathematical morphology strategy	Optimize the number of and locations of indoor antennas , maximize the coverage.	Directed antennas are not taken into consideration
Ko and Gagnon (2015)	3D irregular terrains	3D	Deterministic	A parsing crossover strategy for the genetic algorithm	Maximize the probabilistic coverage on a point and the global coverage	Authors do not prove that the proposed parsing crossover scheme can be a full remedy when GA approach fails
Jiang et al. (2016)	Thermal Sensor Placement in Smart Grid	2D	Stochastic	A genetic approach based on a Gappy proper orthogonal decomposition (GPOD-GA)	Minimize the thermal sensor number	A single objective is considered. No evaluation of the proposed approach using known metrics.
Alia and Al-Ajouri (2017)	No application is given	2D	Deterministic	harmony search-based algorithm	Maximizing the coverage, maximizing the deployed sensor number	A simple model of the network. A bi-objective model: only two objectives are considered. No simulator is used: only Matlab results are presented
Sweidan and Havens (2016)	Terrain-Aware Wireless Sensor Networks	2D	Deterministic	Normalized Genetic approach (NGA), Artificial Immune System (AIS) approach and Particle Swarm Optimization (PSO) based approach	Minimize the cost of mobility and maximize the coverage	The high execution time of the proposed AIS and NGA.
Enayatifar et al. (2014)	A simulation scenario for the deployment of sensors	2D	Deterministic	A new algorithm: objective Imperialist Competitive Algorithm (MOICA).	Minimizing the number of deployed nodes Maximizing the coverage.	No real experimentations The comparison with other approaches is not based on known metrics (such as hypervolume and IGD).
Zainol Abidin et al. (2014)	A simulation scenario for the deployment of sensors	2D	Deterministic	A new algorithm: multi-objective territorial predator scent marking algorithm (MOTPSMA).	Maximizing the coverage and the connectivity. Minimizing the consumed energy.	No real experimentations (using testbeds), even the simulations are performed using Matlab, not a dedicated simulator for wireless networks. No comparison of the new approach with other known metaheuristics.
Cao et al. (2018)	monitoring maritime environments	3D	3D uncertain coverage model	distributed parallel cooperative co-evolutionary multi-objective largescale evolutionary algorithm	Maximizing coverage, lifetime and reliability	the parallelism of the proposed algorithm can be further improved through implementation on GPUs or MICs.
Liu and Ouyang (2018)	Heterogeneous Camera Sensor	3D	probabilistic sector-disk	A mathematical k-coverage estimation	Maximizing k-coverage	A solid modeling but no real expermiments

	Networks		region	expression		
Khalfallah et al. (2017)	3D Underwater WSN Deployment in Rivers	3D	probabilistic	a novel deployment heuristic (3D-UWSN-Deploy) based on subcube tessellation and a mixed ILP optimization	Maximizing Full-Coverage Connectivity, and Quality of Monitoring	The choice of the values of simulation parameters is not justified. Performance metrics are not the standards known ones. Comparison with exact methods.
Brown et al. (2016)	Wireless Video Sensor Network	3D	Deterministic	a greedy heuristic and an enhanced Depth First Search (DFS) algorithm	Maximizing the 3D indoor space coverage	No comparison with other methods Only one objective is considered (the coverage) Video security is not considered
Wu and Wang (2017)		3D	probabilistic sensing	An approximation greedy discretization approach	Maximizing the k-coverage and connectivity	Simplistic Problem Formulation Small-scale problem Instances Comparison with the standard GA

Among the main drawbacks of these studies, the non-consideration of the many-objective case of the problem. Even more, the majority of these studies did not test the proposed approaches on real-world problems which are more complicated. Hence the proposed contributions in this thesis.

1.12 Conclusion

In this chapter, we presented the issues of three-dimensional deployment and its different types, objectives, models and applications. A set of problems which are similar to ours was detailed. Then recent research works dealing with the problem of deployment were identified and criticized. The next chapter will describe the different approaches used to solve the three-dimensional deployment, especially the evolutionary optimization algorithms, the multi-agent systems and the concepts of incorporation of dimensionality reduction and user preferences.

Chapter 2
Optimization methods and muli-agent systems for solving many-objective problems

2.1 Introduction

In the real world, especially in industrial fields such as telecommunication, chemistry, transportation, and mechanics, the problems are complex and have a large scale. This makes them difficult to solve, although they are often easy to define. Optimization, which represents a crucial part of operational research deals with this type of problems. Indeed, most of these problems belong to the class of NP-hard problems and so do not have any efficient and valid algorithmic solutions for all data. In the literature, several studies were interested in solving optimization problems, especially in industrial fields (Khemiri et al., 2017), (Kadri et al., 2014). Optimization problems are divided into two groups: single-objective problems and multi-objective ones. Single objective problems rarely exist in real-world applications. However, multi-objective problems; generally characterized by different conflictual criteria to be simultaneously satisfied; represent a majority of real situations. Whether it is mono-objective or multi-objective, an optimization problem can be classified into one of two categories: a problem having a real-time variable solution also called continuous optimization problem or a problem with a discrete variable solution. Combinatorial optimization is an intermediate type between continuous and discrete optimization. When the problem is multi-objective, it is named multi-objective combinatorial optimization. In what follows in this chapter, we present the different types of combinatory optimization, multi-objective optimization and its resolution methods. Then, we detail MAS and other processes such as user preferences and dimensionality reduction.

2.2 Combinatorial optimization

A combinatorial optimization problem is defined by a set of instances where each instance associates a discrete set of solutions S composed by a subset of admissible (feasible) solutions X and a cost function f (objective function) that associates with each solution s∈X a real number f(s) representing its cost. Solving a combinatorial optimization problem involves finding a solution s* belonging to X to optimize the cost function f. s* is called an optimal solution or global optimum. (Papadimitriou and Steiglitz, 1982) proposed the following definition for a combinatorial optimization problem: An instance I of a minimization problem is a pair (X , f) where $X \subset S$ is a finite set of solutions eligible and f a cost (or objective) function to be minimized. The problem is to find $s^* \in X$ such that $f(s^*) \leq f(s)\ \forall\ s \in X$. Similarly, the problem of maximization can be defined by changing ≤ by ≥.

2.2.1 Complexity of a problem

The complexity of an algorithm is the necessary time and memory space for its execution. It is computed according to the number N. N is the number of data also called the size of the problem (Carlier and Pinson, 1989). In the worst case, the complexity theory increases the number of operations required by a function having the size N of the problem. The complexity of an algorithm is $f(N)$ denoted $O(f(N))$, if it ∃ a constant C and an integer A, such that the number of elementary instructions $I(N)$ satisfy $I(N) \leq Cf(N)$ for all $N \geq A$. If $f(N)$ is polynomial then the algorithm is polynomial (Farreny and Ghallab, 1987).

2.2.2 Classification of optimization problems

Independently from the degree of difficulty of the optimization problem, its resolution requires, first of all, its classification with respect to existing problems in the literature. This classification is not just a way to show the importance of the problem but also to facilitate its resolution by making analogies compared to another existing problem for example. The industrial domain is full of different types of problems which can be grouped into two main classes: The decision problems characterized by a solution reduced to a simple "yes" or "no" answer, and the optimization problems with a permissible solution that must be constructed by minimizing or maximizing the objective function. Another particular type of problem,

called undecidable problems, exists. It is the most difficult to resolve and generally with no method of resolution. Although this classification shows independence between these two types of problems, the literature shows an active link between decision problems and optimization problems. Solving a problem which aims to minimize or maximize the value of the objective function, requires indirectly the resolution of a decision problem which consists in studying the existence or not of an optimal admissible solution. Inversely, the study of the complexity of a decision problem allows giving indications of the associated optimization problems (Charon et al., 1996). Generally, any decision or optimization problem must belong to one of the following three classes:

* **P-Class problems**: Any problem belonging to this class must be a decision problem and should have at least one polynomial algorithm as a method of resolution. This class is named easy or P-class. The verification of the existence of such an algorithm is mandatory to show the belonging of the problem to this class.

* **NP-class problem**: The problems belonging to this type of class, called NP difficult problems, cannot be solved in a polynomial time. The resolution methods used are the heuristics that allow to find an optimal solution but not demonstrable. P-class problems can be solved by conventional or exact algorithms that are included in the family of heuristics. Subsequently, the P class is included in the NP class.

* **NP-Complete Class Problem**: A decision problem P is said NP-complete if it belongs to the class NP and if we have the following properties for any problem P' of NP:

- There is a polynomial application that transforms any instance I' of P' into an instance I of P.
- P' admits a 'yes' answer for the instance I', if and only if P admits a 'yes' answer for the instance I.

A general property of complete NP problems is as follows: If only one NP-complete problem is polynomial, then all NP-complete problems are polynomial. If the problem is NP-hard, there is no polynomial algorithm solving it (Esquirol and Lopez, 2001).

2.3 Multi-objective optimization

The origin of multi-objective optimization dates back to the 19th century. Firstly, it has been applied in the fields of management and economy by Edgeworth and Pareto in the 1980s; then it was extended to engineering sciences in the 1990s. The researchers were initially interested in bi-objective issues. To solve this type of problem, they used methods such as Branch and Bound (Sen et al., 1988), A* algorithm (Stewart and White, 1991) and linear programming (White D., 1982). Different studies have shown the effectiveness of these methods for small size problems, but also their inefficiencies for large size or multicriteria problems. The complexity of the latter type of problems becomes greater. Given the lack of adequate and efficient methods to resolve these problems, different meta-heuristics have been proposed as approaches. Unlike single-objective optimization, which consists of finding a unique solution, multi-objective optimization allows finding a set of solutions that represents all the Pareto optimal solutions.

Définition: An optimization problem consists of searching for the maximum or minimum value, called global optimum, of a function $F: S \rightarrow R$ called the *objective function*. For a multi-objective problem, we minimize or maximize a vector of functions where each component represents an objective function. A multi-objective problem can be defined as follows:

$$\text{PMO} \begin{cases} \min F(x) = (f_1(x), f_2(x), \ldots, f_n(x)) \text{ , } n \geq 2 \text{ is the number of objective functions} \\ s.c\ x \in C \end{cases}$$

$x = \{x_1 , x_2 , ..., x_k \}$ is the vector representing the decision variables, C represents the set of feasible solutions associated with constraints of equality, inequality and explicit bounds

(decision space). $F(x) = (f_1(x), f_2(x), ..., f_n(x))$ is the vector of criteria to be optimized (in minimization or in maximization). The solution of a multi-objective optimization problem is a set of non-dominated solutions known as the set of optimal Pareto solutions (PO).

2.3.1 Pareto Optimality Concepts

*** Notion of dominance**

Let x be a potential solution to the multi-objective problem, $x \in A$ dominates $x' \in A$ *if, and only if,* $\forall i\ f_i(x) \leq f_i(x')$ with at least i such that $f_i(x) \leq f_i(x')$. A solution x is said weakly not dominated, if there is no solution $x' \in A$ *such that* $f_i(x') \leq f_i(x)$ with at least i such that $f_i(x) < f_i(x)$ *where* $i = 1, ..., n$ is the number of objectives. Figure 2.1 illustrates examples of dominance.

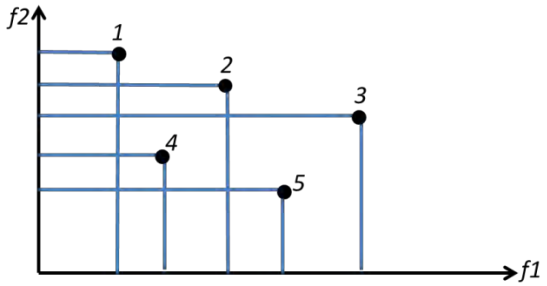


Figure 2.1 Example of dominance

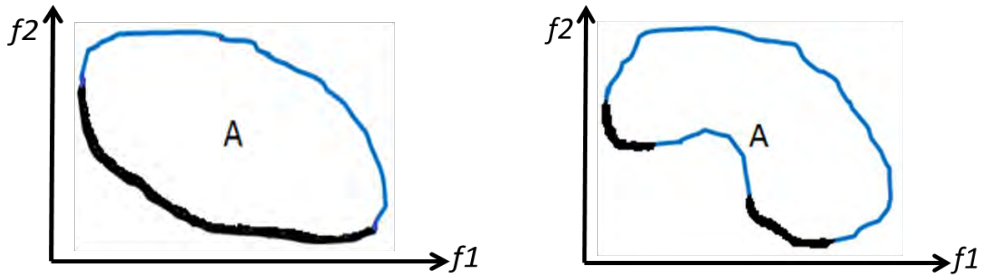
The functions f_1 and f_2 are two functions to be minimized; solutions 1, 4 and 5 are not dominated by any other solution while solutions 2 and 3 are dominated by solution 4.

*** Concept of optimality**

A solution $x^* \in C$ is Pareto optimal (or effective, or not dominated) if and only if, there is no solution $x \in C$ such that x dominates x^*. A solution $y = (y_1, y_2, ..., y_k)$ is said to dominate a solution $z = (z_1, z_2, ..., z_k)$ in the case of a minimization of objectives *if, and only if* $\forall i \in [1..n]$, $f_i(y) \leq f_i(z)$ et $\exists\ i \in [1..n]$ such that $f_i(y) \leq f_i(z)$. Thus, any solution of the Pareto set can be considered optimal since no improvement cannot be made on one objective without degrading the relative value of another objective. These solutions compose the Pareto front.

*** Pareto Front**

The Pareto front is the set of optimal Pareto points. For the previous example in Figure 2.1, the Pareto front is composed of the points 1, 4 and 5. Figure 2.2 illustates an example of a Pareto front for a bi-objective problem. A is the feasible domain.



convex Pareto front non-convex Pareto front

Figure 2.2 Pareto front for a bi-objective problem

*** Concept of dominance with constraints**

A solution ***i*** dominate a solution ***j*** when one of the following situations occurs:

- The solution ***i*** is feasible while the solution ***j*** is not.
- Both solutions are possible and the solution ***i*** dominates the solution ***j***.

- The two solutions are not feasible, but the solution ***i*** has a number of violations of the constraints lower than that of ***j***.

*** Optimum in the sense of Pareto**

In a multi-objective problem, there is a balance which consists in saying that we cannot improve a criterion without damaging at least one of the others. This balance is called the Pareto optimum. A solution *x* is called Pareto optimal if it is not dominated by any other solution belonging to the search space. Optimal Pareto solutions are called non-dominated solutions or non-inferior solutions.

2.3.2 Approaches to solve multi-objective optimization problems

In order to solve a multi-objective optimization problem, a set of Pareto solutions must be determined. However, the resolution requires, as a first step, the choice of the right method. A first classification consists in grouping the resolution methods in three families according to the preferences of the DM and the degree and time of his intervention: *a priori*, *progressive* and *a posteriori* optimization method. Another more interesting classification of the multi-objective optimization problem is based on three classes as follows: the first class concerns approaches that are based on the transformation of a multi-objective problem into a single-objective problem, the second class is based on non-Pareto approaches and the third-class concerns Pareto approaches. Figure 2.5, shown below, presents a classification of the different methods and approaches solving multi-objective problems (Talbi, 1999).

2.3.2.1 Transformation of a multi-objective problem into a single objective problem

This approach is based on the transformation of a multi-objective problem into a single-objective problem. Among these methods of resolution: the aggregation method, the ϵ-constraints method and the goal programming method (global programming) (Hwang and Masud, 1979).

a. Aggregation method: This is one of the first methods used to solve multi-objective problems. It consists in lineary combining several criteria *fi* of the problem into a single criterion *F*. $F(x)=\sum_{i=1}^{n} a_i f_i(x)$ knowing that $a_i \geq 0$ are the weights assigned to the criteria, *x* is the parameter vector of the objective function $\sum_{i=1}^{n} a_i = 1$ and *n* is the number of criteria. The values of α_i will be the components of the vector weight which will be represented by the hyperplane in the space objective (a straight line for a bi-objective problem) with a fixed orientation. The optimal Pareto solution is the point where the hyperplane has a common tangent with the realizable space (the point x in Figure 2.3).

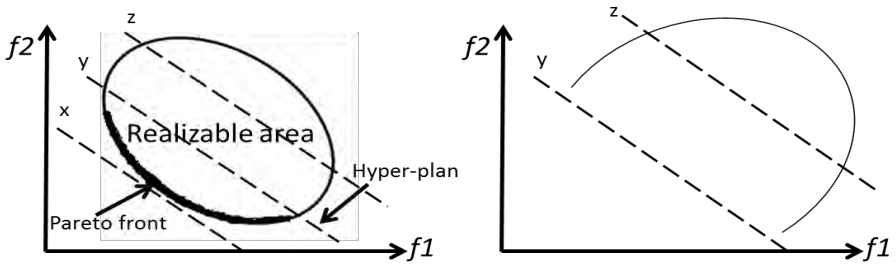


Figure 2.3 Pareto Solutions

The choice of α_i values is crutial because it strongly influences the problem resolution results. This step is sensible because these values depend on the decision maker's (DM) preferences. One of the approaches to have good values of α_i is to solve the problem with several α_i values.

Several "Blind" strategies can be used to generate the weights. In (Veldhuizen et al. 1997), weights are randomly generated: w_i = (random$_i$/ random$_1$+random$_i$+…+random$_n$), i=1, 2, … n where random$_i$ variables are positive integers. The different objectives are generally evaluated in different scales. Standardization of the objectives values is necessary to have an equation with variables in the same scale. Thus, F becomes as follows: $F(x)=\sum_{i=1}^{n} c_i a_i f_i(x)$ where c_i are constants that put the different objectives on the same scale. c_i are usually initialized to $1/f_i(x^*)$, where $f_i(x^*)$ is the optimal solution associated with the objective function fi. In this case, the vector is normalized with respect to the ideal vector.

Although the found solutions may not be acceptable especially if the space of research is prematurely reduced, this approach has some advantages such as determining a single solution and the no need to interact with the DM (except for weight values requiring the knowledge of the addressed problem by the DM).

b. ε-constraint method: The ε-constraint method is based on the optimization of an objective function f_k whose constraints are in function of the constraints of the other functions i # j.

$$\begin{cases} \min f_k(x) \\ x \in C \quad où\ \varepsilon = (\varepsilon_{1,\dots,}\varepsilon_{k-1},\varepsilon_{k+1,\dots,}\varepsilon_n) \\ s.c\ \ f_j(x) \le \varepsilon_j,\ j=1,\dots n,\ j \# k \end{cases}$$

Several optimal Pareto solutions can be generated by modifying the values of the variable ε_i. This requires a priori knowledge of the appropriate intervals for the ε_i values for all objectives. The ideal vector must be calculated to determine the lower bounds. Hence we have: $\varepsilon_i \geq f_i(x^*)$, $i = 1, 2, k-1, k+1, n$.

The example in Figure 2.4 shows a comparison between the original objective space F(C) of a bi-maximization problem and the restricted space F(C') by the ε–constraint transformation.The objective functions f_i can be classified in order of priority following the lexicographic order of the indices of these functions. Therefore, the function f_1 is more priority than f_2. This approach has been applied to several meta-heuristics such as genetic algorithm (Ritzel et al., 1994), (Hertz et al., 1994) and taboo research (Wienke et al., 1992). In general, this method generates weak Pareto optimal solutions. However, if the optimal solution is unique, then the found solution becomes strongly Pareto optimal. Generating multiple solutions requires running the algorithm multiple times (with different constraints), which is high-cost computing time.

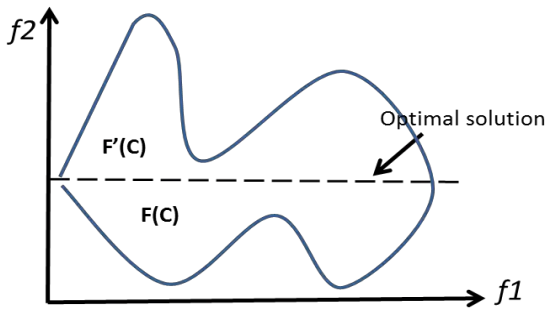


Figure 2.4 ε–contsraint method

c. Goal Programming Method: It consists in defining the goals presented by a priori values for the objectives to be achieved. These values will be used to transform the multi-objective problem into a single-objective one.

$$\begin{cases} \min \left(\sum_{j=1}^{n} \lambda_j \mid f_j(x) - z_j \mid^p\right)^{1/p} \\ \text{s.c } x \in C \end{cases}$$

Where z is the reference vector (goal) or the ideal vector and $1 \leq p \leq \infty$. The used norm is the metric of Chebysheff (Lp-metric). Generally, if p is equal to 2, we have a Euclidean metric. If $p = \infty$, the equation come back to a Min-Max function. An arbitrary selection of the reference vector may be a wrong choice of the reference vector because it can lead to a solution that is not Pareto optimal. This approach has been applied to several meta-heuristics such as genetic algorithms (Coello, 1998) (Serafini, 1992). Simulated Annealing (Gandibleux et al., 1996) and taboo search (Collette and Siarry, 2002).
Figure 2.5 shows a classification of multi-objective optimization methods.

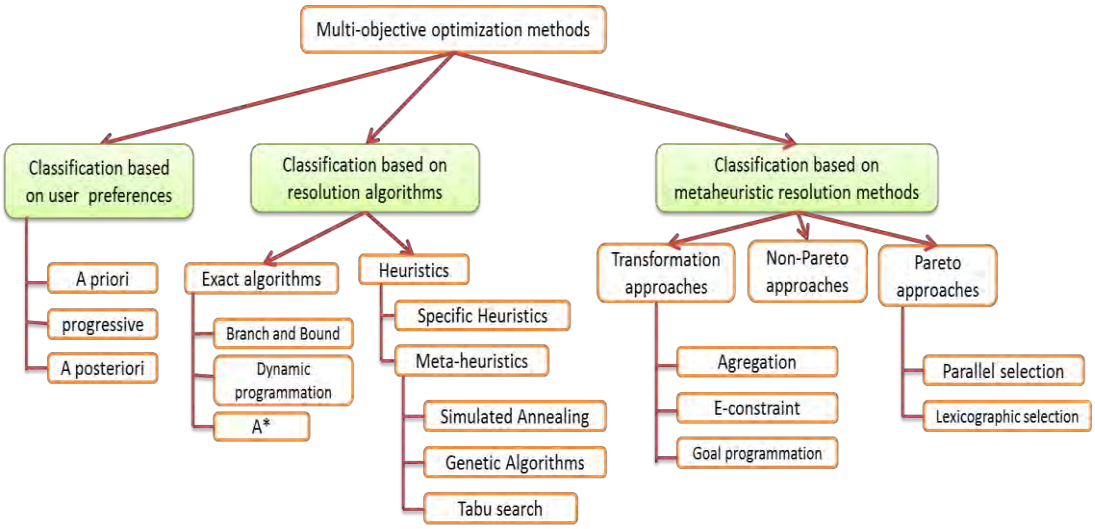


Figure 2.5 Classification of multi-objective optimization methods (Talbi, 1999)

2.3.2.2 Non-Pareto approaches (non-aggregated)

Methods based on this approach treat the objectives separately. We can distinguish three groups of processes:

a. Lexicographic selection: The selection is carried out using an order which is defined a priori by the DM according to the importance of the objectives. This order allows determining the weight of the objectives. Many heuristics are based on this approach such as the genetic algorithm. A lexicographical approach was proposed in (Kursawe, 1991). This approach compares the individuals according to a randomly chosen objective based on a predetermined probability.

b. Parallel selection: The VEGA method proposed in (Schaffer, 1985) consists in selecting the Pareto Front solutions following each objective independently from other objectives. The principle of the algorithm consists of dividing the population into *n* subpopulations knowing that *n* is the number of objectives. Each subpopulation *i* is selected according to the objective *i*. the analysis of this algorithm showed that its behavior is the same as an algorithm performing a linear aggregation (Richardson et al., 1989).

c. Multi-Sexual Reproduction: As proposed in (Allenson, 1992), the principle of this method is to associate with each class of individuals an objective using a bijective function. Each class of individuals is identified by a different gender. An individual of a given gender is evaluated according to the associated objective function of its gender. At the creation of an individual, its gender is randomly choosen and assigned. An individual can to be of male or

female. Reproduction can only be between a male individual and a female individual. At the initialization of the population, the number of male and female indi viduals is equal. For the next generations, Evolutionary algorithms can be used to implement a form of sexual attractors. This method allows defining subpopulations separated by objectives. Parallel selection and lexicographic selection differ in the multi-sexual reproduction by the neighborhood restriction. This latter is used in the *lexicographic selection* to avoid the random reproduction of individuals.

2.3.2.3 Pareto approach

Unlike other approaches combining criteria and treating them separately, the Pareto approach uses the concpet of dominance to select solutions that allow the population to converge towards a set of effecient solutions. These methods allow having a set of possible Pareto solutions of the problem, but do not allow comparing solutions. The choice of the final solution is decided by the user (DM).

2.4 Methods for solving multi-objective optimization problems

The choice of a resolution method for a multi-objective problem necessarily depends on the complexity of the problem as well as its size. A simple small-size optimization problem can be solved using an exact method having a reasonable execution time. However, for difficult large-size problems, the exact methods are not efficient. Then, meta-heuristics become the most appropriate. Although the meta-heuristic is approximate (and not optimum), its execution time is reasonable and favors its choice for difficult and complex problems.

2.4.1 Exact methods

For large size problems, the calculation time becomes unreasonable. Exact methods provide optimal solutions only for small size problems. Among these methods, the Branch and Bound which is the most used.

*** Branch and Bound**

This method provides solutions by exploring a search tree describing all the possible solutions. It consists of finding the best configuration to avoid (to prune) the branches of the tree that lead to inadequate solutions. To find the best solution, the Branch and Bound algorithm performs a full search in a given problem space. The steps of the Branch and Bound algorithm are as follows:

1. Divide the search space into a set of subspaces.
2. Find a minimum bound for each objective function associated with each sub-search space.
3. Eliminate bad spaces.
4. Reproduce the previous steps until reaching the global optimum.

(Lin, 2015) suggests a branch and bound algorithm to resolve the problem of deployment of smart gateway in smart home environments. This study investigates the connections between the different devices in the network and its applications in the systems of home automation. The aim is to minimize the cost of deployment while achieving the coverage of all service areas. However, the size of the problem is small and is not suitable for real contexts.

*** Dynamic programming**

It is based on the principle of Bellman (Bellman, 1986): "*if a point C belongs to the optimal path between A and B, then the portion of this same path from A to C is the optimal path between A and C*". Therefore, this method consists in constructing the optimal subpaths then recursively constructing the optimal path for the entire problem.

(Li et al, 2015) introduce a strategy, named EDSNDA for deploying nodes in WSN. This strategy relies on a dynamic programming model to optimize the coverage, the connecivity and the number of needed nodes. Althought its efficiency compared to other mecthds (Max_cov and Min_cov), the given results are based only on simulations.

2.4.2 Meta-heuristics

Meta-heuristics are NP-complete methods used to solve combinatorial optimization problems. Several definitions have been proposed. (Osman and Laporte, 1996) introduced the following description : "It is an iterative process that subordinates and guides a heuristic by cleverly combining several concepts to explore and exploit the entire research space. Learning strategies are used to structure information in order to find optimal solutions or almost optimal ones" (Osman and Laporte, 1996). The fundamental properties of meta-heuristics are as follows:

• They are strategies to guide the research and effectively explore the space in order to achieve a solution close to optimal.
• They are non-deterministic since the optimal solution is not guaranteed.
• They can range from a simple local search procedure to a complex learning process (Widmer et al., 2001).

2.4.2.1 Taboo Search (TS) / Simulated Annealing (SA)

*** Simulated annealing**: proposed by (Kirkpatrick et al., 1983), it is a method inspired by a physical procedure which is used by metallurgists. To have a flawless alloy, we heat the metal and let it cool very slowly (annealing technique) until reaching its thermodynamic equilibrium. This method exploits the algorithm in Algorithm 2.1 which consists in introducing an elementary modification to the objective function F. If this modification decreases the value of this function, the solution will be accepted. Otherwise, it will be recognized with a probability of exp (ΔF/T). Note that the choice of the temperature function is decisive.

01: Variable : s (current solution)
02: Begin
03: T ← T0
04: Choose an initial solution s_0 ;
05: Current solution s ← s_0 ;
06: Repeat
07: Randomly choose s'∈N(s)
08: Generate a real random number *r* in [0,1]
09: if $r < e^{\frac{f(s)-f(s')}{T}}$ **then** s ← s'
10: Update T
11: Until the termination criterion is met
12: End

Algorithm 2.1 Simulated Annealing algorithm

The authors in (Ateş et al, 2017) introduce a priority-estimation approach for WSN deployment based on area-priority. A K-means-clustered satellite image of the environment is used to identify the coverage-areas priorities of the sensor coverage areas on the image positions. After a priority-queue-based initial deployment, a simulated annealing method is achieved to minimise gaps between sensors and maximise the covered area priority. Although different experiments prouve the efficiency of the suggested approach cmpared to the random deployment, there is no comparaison with other optimisation algorithms.

*** Taboo search:** Proposed by Fred Glover in the 80s (Glover, 1989), it is a classical combinatorial optimization method. Its main feature is a history of visited solutions allowing a minimum blind search. Using this history, the Taboo search is able to avoid periodically falling in the same local optima. The Tabou method examines a solution sampling of N (s)

and retains the best solution s' even if f(s') > f(s). Thus, the taboo search does not stop at the first found optimum. To avoid the risk of immediately return to s, since s is better than s', the last visited solutions will be stored in a Taboo list and forbid any move to any item on this list. The Taboo search algorithm is illustrated in Algorithm 2.7, knowing that $N_T(s) = \{s'\}\ s' \in N(s)$.

01: Begin
02: Choose an initial solution s
03: Let T←Φ and s'←s ;
04: Repeat
05: Choose s' that minimize f(s') in $N_T(s)$
06: if f(s')<f(s*) **then** s*←s' ;
07: Let s←s' and and update T
08: Until the termination criterion is met
09: End

Algorithm 2.2 Tabu Search algorithm

One of the most interesting studies using a Tabu Search method to solve the deployment of sensors in WSN, was proposed in (Aitsaadi et al., 2008). Authors formulate the problem as a multi-objective one, aiming to guarantee a minimum detection probability and number od used nodes. Compared to random, grid, MIN MISS, MAX MIN COV, and MAX AVG COV algorithms, the results of the Tabu Search algorithm are encouraging.

2.4.2.2 Swarm Intelligence (ACO, PSO, BSO)

**** Ant colony optimization (ACO)***

ACO algorithms are inspired by the behavior of ants proposed by Marco Dorigo et al in the 90s Dorigo and Di Caro (1999). In nature, an ant colony having the choice between two paths of unequal length leading to a food source, tend to use the shortest route. The model explaining this behavior is as follows:

1. An ant called "scout" walks randomly around the colony.
2. If this ant discovers a food source, it returns to the nest, leaving a trail of pheromones on its path.
3. These pheromones are attractive, ants pass nearby will tend to follow this track.
4. Returning to the nest, these same ants will strengthen the trail.
5. To reach the same food source, if two tracks are possible, the shortest one will be traveled by the most ants.
6. The short path will be stronger and therefore more attractive
7. Sine the pheromones are volatile, the most extended path will eventually disappear.

The ACO algorithm was introduced by (Dorigo et al., 1996) in order to resolve the problems of hard CO (Combinatorial Optimization). It is a bio-inspired approximate algorithm which aims to obtain good solutions with reasonable computational cost (time) when resolving CO problems. Moreover, it is a meta-heuristic which is considered as probabilistic and population-based. The ACO stems from the foraging behavior of the real ants. Indeed, the ants aim at finding the shortest path between its nest and the source of food. Instead of using visual informations, ants use a chemical substance named 'pheromone' which is left behind their trails. So, in the ACO, the artificial ants (called agents) imitate their natural counterparts to resolve the problems by finding the optimal solutions. The ACO algorithm is illustrated in Algorithm 2.3. Indeed, firstly, when the collection begins, the shortest route leading to food is not known. Thus, each ant pursues a route randomly and place pheromone. As the collection progress, ants continue putting the pheromone. As a consequence, all traveled routes contain this substance. Then, because the pheromone evaporates overtime, if an ant wants to travel, it chooses the route with the highest rate of pheromone, which corresponds to the shortest among all routes. Therefore, overtime, only one route will remain (the shortest one).

```
01: Initialize the ACO parameters (pheromone, ..)
02: Each_ant_builds_a_solution()
03: Evaluate_the_solutions
04: Initialize the number of travels per ant, t = 1
05: While t < lt_max do
06:    Update_the_pheromone()
07:    Each_ant_builds_a_ new_solution()
08:    Evaluate_the_solutions
09:    t = t + 1
10: End while
```

Algorithm 2.3 Ant Colony Optimization algorithm

In (Deif and Gadallah, 2017), authors adress the minimum cost reliability constrained deployment problem (MCRC-SDP) using an hybrid approach based on ant colony and local search. The experimental results relying on a comparaison with the greedy heuristic show the efficiency of the suggested methos.

Particle swarm optimization algorithm (PSO)

General Principle: PSO is an evolutionary algorithm that uses a population of candidate solutions to develop an optimal solution to the problem. This algorithm was proposed by Russell Eberhart (an electrical engineer) and James Kennedy (a socio-psychologist) in 1995 (Kennedy and Eberhart, 1995). It is inspired by the origins of the world of living beings, specifically the social behavior of swarming animals, such as fishes and birds. We can observe the complex and dynamic displacement of these animals where each, if does not interact with others, has a limited "intelligence" and local knowledge of his situation in the swarm. The local information and the memory of each are used to decide its movements. Simple rules, such as "staying close to other people", "going in the same direction" or "going to same speed", are sufficient to maintain the cohesion of the swarm, and allow the implementation of complex and adaptive collective behaviors. The swarm of particles corresponds to a population of simple agents, called particles. Each particle is considered as a solution to the problem containing a position (the solution vector) and a velocity. In addition, each particle has a memory allowing it to remember its best performance (in position and value) and the best performance achieved by the "neighboring" (informative) particles: each particle has a group of informants, historically called his neighborhood. A swarm of particles composed of the potential solutions of the problem, "flies" over the search space in search of the global optimum. The displacement of a particle is influenced by the following three components:

1. A component of inertia: the particle tends to support its current direction of mouvement.
2. A cognitive component: the particle tends to move towards the best site by which it has already passed.
3. A social component: the particle tends to rely on the experience of its congeners and move towards the best site already reached by its neighbors.

The strategy for moving a particle is illustrated in Figure 2.6.

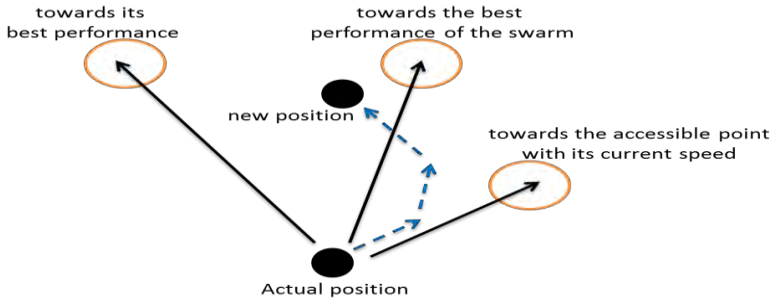


Figure 2.6 Displacement of a particle [inspired from (Eldor, 2012)]

Formalization: In a research space of dimension D, the particle i of the swarm is modeled by its position vector $\vec{x}_i = (x_{i1}, x_{i2}, ..., x_{iD})$ and velocity vector $\vec{v}_i = (v_{i1}, v_{i2}, ..., v_{iD})$. The quality of its position is determined by the value of the objective function at this point. This particle keeps in memory the best position by which it has already passed, noted $\vec{P}best_i = (pbest_{i1}, pbest_{i2}, ..., pbest_{iD})$. The best position reached by the particles of the swarm is noted by $\vec{G}best = (gbest_1, gbest_2, ..., gbest_D)$. We refer to the global version of PSO, where all the particles of the swarm are considered as close to the particle *i*, hence the notation $\vec{G}best$ (global best).

The term "speed" is abusive here because $\vec{v}_i$ vectors are not homogeneous at one speed. It would be more appropriate to use the term of "direction of displacement". However, to respect the analogy with the animal world, the authors preferred to use the term "speed". At the beginning of the algorithm, the particles of the swarm are randomly and regularly initialized in the search space of the problem. Then, at each iteration, each particle moves and linearly combines the three components mentioned above. Indeed, at the iteration t+1, the velocity and position vectors are calculated using equation (2.1) and equation (2.2), respectively.

$$v_{i,j}^{t+1} = wv_{i,j}^{t} + c_1 r_{1_{i,j}}^{t} \left[pbest_{i,j}^{t} - x_{i,j}^{t} \right] + c_2 r_{2_{i,j}}^{t} \left[gbest_{j}^{t} - x_{i,j}^{t} \right], j \in \{1, 2, ..., D\} \quad (2.1)$$

$$x_{i,j}^{t+1} = x_{i,j}^{t} + v_{i,j}^{t+1}, j \in \{1, 2, ..., D\} \quad (2.2)$$

where w is a constant, named the coefficient of inertia; c1 and c2 are two constants, named acceleration coefficients; r1 and r2 are two random numbers uniformly taken in [0, 1] at each iteration t for each dimension j.

The three components mentioned above (i.e. of inertia, cognitive and social) are represented in equation (2.1) by the following terms:

1. $wv_{i,j}^{t}$ corresponds to the component of inertia of the displacement, where the parameter *w* controls the influence of the direction of the displacement on the future displacement;
2. $c_1 r_{1_{i,j}}^{t} \left[pbest_{i,j}^{t} - x_{i,j}^{t} \right]$ corresponds to the cognitive component of displacement, where the parameter c1 controls the cognitive behavior of the particle;
3. $c_2 r_{2_{i,j}}^{t} \left[gbest_{j}^{t} - x_{i,j}^{t} \right]$ corresponds to the social component of displacement, where the parameter c2 controls the social aptitude of the particle.

Once the displacement of the particles has been carried out, the new positions are evaluated and the two vectors $\vec{P}best_i$ and $\vec{G}best$ are updated, at the iteration t+1, according to the two equations (2.3) (in the case of a minimization) and (2.4) (in the global version of the PSO), respectively.

This procedure is presented in Algorithm 2.4, where N is the number of particles in the swarm.

$$\vec{P}best_i(t+1) = \begin{cases} \vec{P}best_i(t), & if\ f(\vec{x}_i(t+1)) \geq \vec{P}best_i(t) \\ \vec{x}_i(t+1), & otherwise \end{cases} \quad (2.3)$$

$$\vec{G}best(t+1) = \arg\min_{\vec{P}best_i} f(\vec{P}best_i(t+1)),\ 1 \leq i \leq N \quad (2.4)$$

01: Randomly initializes the position and speed of N particles.
02: Evaluate the particle positions
03: For each particle *i*, $\vec{P}best_i = \vec{x}_i$
04: Calculate $\vec{G}best$ according to (2.4)

05: While the stopping criterion is not satisfied **do**
06: Move the particles according to (2.1) and (2.2)
07: Evaluate particle positions
08: Update $\vec{P}best_i$ and $\vec{G}best$ according to (2.3) and (2.4)
09: End

Algorithm 2.4 PSO algorithm

In (Chen and Jiang 2016), authors propose a Particle Swarm method, named PSCD, to resolve the charger deployment problem in Wireless Rechargeable Sensor Networks (WRSNs). PSCD uses the angle and distance between nodes and chargers to estimate the charging efficiency. Afterwards, to sustain WRSNs lifetime, PSCD uses the local and global optimal result to adjust charger's antenna orientations and positions. PSCD is tested using simulation than compared with two other heuristic greedy algorithms prouve its efficiency.

**** Bees Colonies Optimization (BCO)***

In this algorithm, the location of the food source represents the possible solution of the problem and the quantity of the food of this source corresponds to an objective value called *fitness*. Foragers are assigned to different food sources to maximize the total food intake. The colony must optimize the overall efficiency of the collection. Therefore, the distribution of bees is achieved according to many factors such as the amount of food and the distance between the food source and the hive. This problem is similar to the distribution of web hosting servers, which was actually one of the first problems solved using bee algorithms by (Nakrani and Tovey, 2004). The number of active or inactive foragers represents the number of solutions in this population.

In the first step, the algorithm generates an initial population of solutions randomly distributed. Each solution $x_i\ (i = 1, 2, \dots, SN)$ represents a vector of solutions to the optimization problem. The variables that each vector contains must be optimized. After initialization, the population of solutions is subject to repeated cycles $C = 1, 2, \dots, C_{max}$. These cycles represent research processes performed by active, inactive foragers and scouts.

Active foragers search in the vicinity of the previous source x_i for new sources v_i having more food and then they calculate their fitness. In order to produce a new food source from the former, we use the following expression:

$$v_{ij} = x_{ij} + \phi_{ij} \times (x_{ij} - x_{kj})$$

Where $k \in \{1, 2, \dots, BN\}$ (BN is the number of active foragers) and $j \in \{1, 2, \dots, SN\}$ are randomly chosen indices. Although k is randomly determined, it must be different from i. ϕ_{ij} is a random number belonging to the interval [-1, 1], it controls the production of a food source in the neighborhood of x_{ij}.

After the discovery of each new food source v_{ij}, a greedy selection mechanism is adopted and this source is evaluated by the artificial bees, its performance is compared to that of x_{ij}. If the food in this source is equal to or better than that of the previous source, this one is replaced by the new one. Otherwise the old one is kept.

For a minimization problem, the fitness is calculated according to this formula:

$$fit_i(\overrightarrow{x_i}) = \begin{cases} \dfrac{1}{1 + f_i(\overrightarrow{x_i})} & si\ f_i(\overrightarrow{x_i}) \geq 0 \\ 1 + abs\left(f_i(\overrightarrow{x_i})\right) & si\ f_i(\overrightarrow{x_i}) > 0 \end{cases}$$

$f_i(\overrightarrow{x_i})$ is the value of the objective function of the solution $\overrightarrow{x_i}$. At this stage, inactive foragers and scouts are waiting in the hive. At the end of the research process, active foragers share food information from food sources and their locations with other bees via waggling dancing. The latter evaluate this information from all active foragers, and choose the food sources

according to the probability value P_i associated with this source and calculated by the following formula: $P_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n}$. fit_i is the fitness of the solution i which is proportional to the amount of food in the position i. The scouts replace the source of food which is abandoned by bees with a new source. If a position cannot be improved during a predetermined cycle number called "limit", the food source is assumed to be abandoned. All these steps are summarized in the algorithm in Algrithm 2.5.

Input: S (number of foragers), W (number of active foragers), O (number of inactive foragers)
Output: the best solution
01: Initialize the population with S+W random solutions
02: Evaluate the fitness of the population
03: While the stopping criterion is not satisfied **do**
04: Recruit O inactive foragers and assign each to a member of the population
05: **For** each inactive forager assigned to a limb n of the population **do**
06: Iterate the algorithm of the search for a new source
07: **End for**
08: Evaluate the fitness of the population
09: **If** the members of the population have not improved during the iterations **then**
10: Save the solution and replace it with a random solution
11: Find S random solutions and replace the S members of the population having bad fitness
12: End while
13: Return the best solution

Algorithm 2.5 Bee colony optimization algorithm

In (Udgata et al., 2009), the problem of node deployment in WSN is formulated as a data-clustering problem and resolved using an Artificial Bee Colony (ABC) method. Althought the quality of ABC solutions should be compared to other algorithms to proue its performance.

2.4.2.3 Genetic algorithms (NSGA-II as an exemple)

Since the 20th century, the studies of the scientist Charles Darwin affirm that the evolution of species and organisms uses several mechanisms leading to the emergence of new species always better adapted to their respective environments. Among these mechanisms, the selection and the reproduction. Selection is a mechanism which is used to select the most « robust » individuals (chromosomes) in order to reproduce. The reproduction is the mechanism producing descendants that are not identical, it is the evolution phase of the species. As a result, the idea of algorithms named evolutionary algorithms is developed by J.Holland (Holland, 1975). It is based on the principles of natural evolution (coding, selection, crossing and mutation). By analogy with these evolution principles, Holland studies aim to allow the computer to mimic the natural evolutionary mechanisms in order to solve the problems encountered in reality. These studies proposed a first canonical genetic algorithm to solve optimization problems (Bourazza, 2006). Genetic algorithms (GA) are stochastic algorithms belonging to the family of evolutionary algorithms based on the mechanisms of natural evolution and genetics. The operation of a genetic algorithm begins by the selection of a population of potential 'initial solutions' (chromosomes) , generally randomly chosen. The relative performance of each individual is evaluated by a fitness function that determines its quality. The generation of a new population of potential solutions is based on the evolutionary operators: selection, crossing and mutation. This evolutionary cycle is repeated until a stopping condition is met. This condition can be a maximum number of generations or a stability degree of the characteristics of the solution. Actually, GA is in wide use in different domains such as economics, finance, optimal control theory, the theory of games, scheduling or engeneering real-world problems.

a. Principle of the Genetic Algorithm

Genetic algorithms are the combination of two domains: biology and computer science. To define the functioning of a genetic algorithm, we start by setting the technical used words:

- **Genesis:** is the first phase of the algorithm, it is an initial population of size N.
- **Chromosome:** is a chain representing the characteristics of the individual.
- **Phenotype:** is a set of parameters or a decoded structure.
- **Evaluation:** is the calculation phase of the fitness function.
- **Selection:** is the choice of the individuals who will reproduce.
- **Crossing:** is the phase of production of the descendants
- **Mutation:** is the modification of a chromosome in order to improve the characteristics of the individual.

As shown in Figure 2.7, the GA is composed of five steps:
1. Initialization of the situation.
2. Selection for reproduction.
3. Crossing of the selected individuals.
4. Mutation of the selected individuals.
5. Selection for replacement.
6. If the stop condition is satisfied then STOP, S = {the best individuals}
Otherwise go back to step (2).

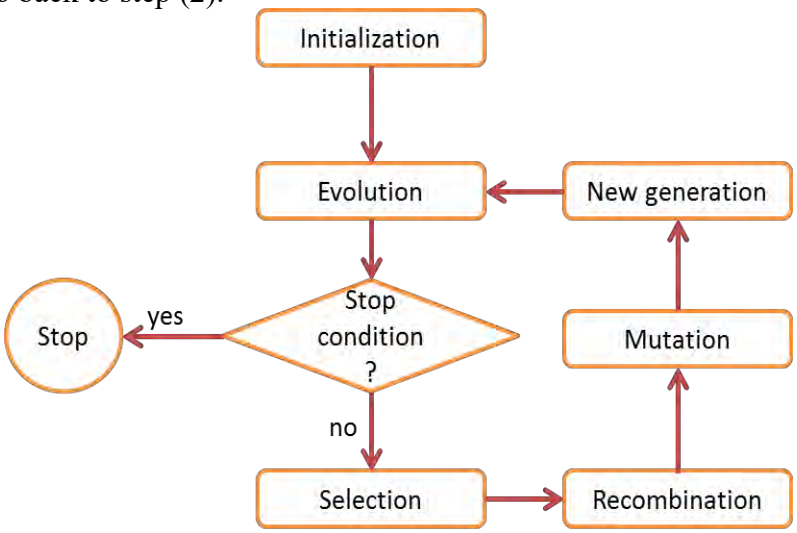


Figure 2.7 General architecture of a genetic algorithm

➢ **Types of chromosome coding**

The use of GA for resolving a given problem requires the coding of its data. This coding is related to the nature of the problem. It is based on establishing a connection between the values of the variable and the individuals of the population in order to imitate the link which exists in biology between the genotype and the phenotype. Three types of coding exist:
- Digital: binary or real, this type is used if the alphabet of the problem consists of digits.
- Symbolic: if the alphabet is a set of alphabetic letters or symbols.
- Alpha-numeric: if the alphabet of the problem is a combination of letters and numbers (Bourazza, 2006).
The correct choice of coding strongly conditions the effectiveness of the genetic algorithm.

b. Individual selection methods

Selection is the choice of individuals for reproduction and mutation. Several selection methods exist in the literature:
* **Sélection by rank**: This method consists in attributing to each individual a classification according to the value of its objective function. For a maximization problem, individuals are ranked in descending order of the values of the objective function. Thus, the worst individual

that is the one with the lowest value of the objective function will take rank 1. For a minimization problem, the ranking will be the opposite of the previous case. The new population is then taken from sets of ordered individuals using probabilities that are indexed according to the ranks of individuals. The probability of selection of $Parent_i$ is equal to $Rang(Parent_i) / \sum_{j \in population} Rang\left(Parent_j\right)$

*** Roulette selection**: This method consists in associating with each individual *i* a probability of selection denoted $Prob_i$ proportional to its objective function value F_i. $Prob_i = F_i / \sum_{j \in population} F_j$. Each individual is then reproduced with a $Prob_i$ probability. Some individuals (the good ones) will be reproduced more and others (the bad ones) eliminated. For a minimization problem, the selection probability used for an individual *i* is equal to (1-prob*i)*/(N-1).

*** Selection by Tournament**: The principle of this method is to randomly choose a subpopulation of size M fixed a priori by the user. Better individuals with respect to the sub-population will be selected for the application of the crossover operator. Indeed, it is a competition between individuals of a subpopulation of size M (M≤N), N is the size of the population. This method gives more chance to poor quality individuals to participate in improving the quality of the population. The M parameter plays an essential role in the tournament method. If M=N, the tournament selection gives each time a single individual that is the best individual according to the value of the objective function, which reduces the genetic algorithm to a local search algorithm. If M = 1, the selection corresponds to the random selection.

c. Crossing operators: Crossing consists of generating two children from two parents with a probability of crossing P_x in order to enrich the diversity of the population. In the following, two crossing operators are detailed.

- One-point operator: It consists in dividing every two parents (P1 and P2) into two parts at the same position, chosen at random. The first child (D1) is composed of two parts of both parents. The first part is that of the first parent and the second one is that of the second parent. The second child (D2) is composed of two parts, the first part of the second parent and the second part of the first parent.

P1	0	1	1	1	0	1	0	1	**D1**	0	1	1	0	1	0	1	0
P2	1	1	0	0	1	0	1	0	**D2**	1	1	0	1	0	1	0	1

- Two-point operator: This method consists in setting two positions. The first child will be the copy of the first parent by replacing its part between the two positions by that of the second parent. The same operation will be applied to determine the second child by reversing the roles of the first and the second parent.

P1	0	1	1	1	0	1	0	1	**D1**	0	1	1	0	1	1	0	1
P2	1	1	0	0	1	0	1	0	**D2**	1	1	0	1	0	0	1	0

d. Mutation operator: In accordance with the natural mutation, this operator changes the value of a chromosome in order to improve the characteristics of the individual. It allows

the GA to efficiently explore the search space. Moreover, it guarantees a susceptibility to reach the majority of points in the realizable domain.

Individual I	0	1	1	0	1	1	0	1

Individual I after mutation	0	1	1	1	1	1	0	1

e. Advantages of GA

-The use of an objective function (fitness) regardless of its nature (convex, continuous or differentiable) which gives it more flexibility and a large field of application.

-The solution is a population of size N, which allows the generation of a form of parallelism.

-The probability of crossing and mutation avoid falling into a local optimum and moving towards the global optimum.

f. Non dominated Sorting Genetic Algorithm II (NSGA-II)

NSGA-II was introduced by Deb and Agawal (Deb et al., 2002) to correct the criticisms of NSGA-I. This algorithm estimates the density of solutions that surround a particular solution in the population by calculating the distance between two points and their sides for all the objectives of the problem. This value is called "crowding distance" during the selection. The algorithm NSGA-II uses the operator "crowded comparison" which takes into account the ranking of non-dominance of the individual in the population and its "crowding distance" (i. e. non-dominated solutions are preferred over dominated solutions). If two solutions have the same rank of non-dominance, the one that resides in the least public region is preferred. The NSGA-II steps are shown in Figure 2.8.

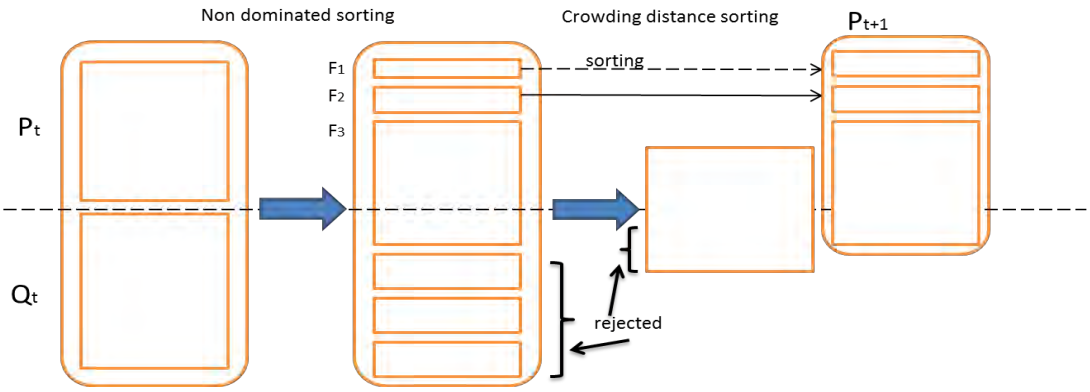


Figure 2.8 NSGA-II Steps

The NSGA-II algorithm is considered in the literature as one of the most efficient algorithms for solving a multi-objective optimization problem using the Pareto approach. However, many-objective optimization problems are more complex and even the NSGA-II becomes inefficient.

2.4.3 Many-objective optimization

2.4.3.1 Many-objective PSO algorithm

Proposed by (Kennedy and Eberhart, 1995), the PSO is a stochastic evolutionary algorithm which is inspired by the social behavior of animals such as fishes and birds. In this algorithm, the particles are initially scattered in the search space in a random manner and they cooperate to achieve an optimal global objective in the Pareto sense. Each particle is characterized by a current position noted $\vec{X}_i(t) = \vec{X}_i(t-1) + \vec{V}_i(t)$ (14) and a speed of movement noted $\vec{V}_i(t) = \omega\ \vec{V}_i(t-1) + c_1 r_1 (\vec{X}_{pbest} - \vec{X}_i(t)) + c_2 r_2 (\vec{X}_{gbest} - \vec{X}_i(t))$ (15) where ω

represents the weight of inertia that controls the speed of change as a function of the current speed. c_1 and c_2 are two knowledge factors. r_1 and r_2 are two random values in [0..1].
A particle can maintain its best-visited position, noted pbest. Moreover, it can access the best position visited by its neighbors, named gbest. The particle changes its position by following these tendencies: A conservative tendency in which the particle tends to return to its best-visited point, an adventurous tendency in which the particle tends to find a better position in the search space and a panurgical tendency in which the particle tends to follow the best position found by its neighbors.
In many-objective optimization problems where the number of objectives (often conflicting) exceeds three, the most important challenging issue is how to obtain a well distributed non-dominated set of solutions which are close to the Pareto Front (PF) in the objective space. Different MaOPSO were proposed in several studies to resolve the lack of diversity and convergence in many-objective problems: In (Figueiredo et al. 2016), the authors proposed a MaOPSO algorithm relying on a set of reference points to recognize the best solutions and guide the search process according to these reference points. The study in (Díaz-Manríquez et al., 2016) proposed an algorithm that empowers the multi-objective structure of the PSO to deal with many-objective problems and suggest a $R2$ indicator to guide the search. In (Britto and Pozo, 2012), the authors proposed a based-on archiving PSO algorithm named I-MOPSO which explores specific aspects of the MOPSO to hundle many-objective problems while introducing more convergence and diversity on the search.
In (Hu et al., 2017), a MaOPSO based on a two-stage strategy and a parallel cell coordinate system is introduced to separately emphasize the diversity and convergence at different stages using a many-objective optimizer and a single-objective one, respectively.

2.4.3.2 NSGA-III algorithm

NSGA-III (Deb and Jain 2014) is a recent algorithm, proposed as an extension of NSGA-II (Deb et al., 2002). It uses a reference point-based approach to solve many objective problems (MaOPs). NSGA-III use the same concept of weight vector generation in MOEA/D (Zhang and Li, 2007) to determine a set of reference points scattered over the objective space. At every generation of each solution, the values of the objective function are normalized to [0, 1]. Then, a reference point is associated with each solution based on its perpendicular distance to the reference line. Assigning a reference point to each solution ensures the uniform repartition of the reference points across the normalized hyper-plane. The generated offspring is combined with the parent to create a hybrid population. Afterward, the hybrid population is divided into a set of non-domination levels according to a non-dominated sorting procedure. The next parents as composed of the solutions in the first level so on and so forth. A niche-preservation operator is used to select solutions in the last acceptable level where the solutions associated with a less crowded reference line are more likely to be selected. For the majority of test problems, NSGA-III which is proposed especially for many-objective optimization shows superior performance compared to other methods such as MOEA/D and NSGA-II. Algorithm 2.6 illustrates the NSGA-III algorithm.

Input: P_0(Initial Population), N_{Pop} size of the population, t (iteration) = 0, It_{max} (Maximum iteration)
Output : P_t
While $t<It_{max}$ **do**
  Create Offspring Q_t
  Mutation and recombination on Q_t
  Set $R_t=P_t \cup Q_t$
  Apply non-dominated sorting on R_t and find $F1$, $F2$, …
  $S_t=\{\}$, $i=1$
  While $|S_t| \leq N_{Pop}$ **do**
    $S_t=S_t \cup F_i$
    $i=i+1$
  End While
  If $|St| <> N_{Pop}$ **do**
    $P_{t+1}= \cup_{j=1}^{i-1} F_j$

13: *Normalize* S_t using min and intercept points of each objective
14: *Associate* each member of S_t to a reference point
15: *Choose* $N_{Pop}-|P_t+1|$ members from F_l by te niche-preserving operator
16: **Else** $P_{t+1}=S_t$
17: **End if**
18: $t= t+1$
19: End While

Algorithm 2.6 The NSGA-III algorithm

2.4.3.3 MOEA/DD

This algorithm (Li et al. 2015) aims to achieve the balance between diversity and convergence which is a key issue in resolving scalable multi-objective optimization. Most of the existing methodologies, demonstrating efficiency in solving real-world problems with two and three objectives, face significant challenges in optimizing problems with more than four objectives. MOEA/DD suggests a unified paradigm, which combines approaches based on decomposition and dominance for multi-objective optimization. The primary goal is to balance the convergence and diversity of the evolutionary process using the decomposition and dominance metrics. Compared to other recent constrained optimizers, the MOEA/DD is very competitive on constrained optimization problems with up to 15 objectives. On the other hand, as the performance of an optimization algorithm may degenerate with the increase in the number of decision variables, it is useful to study the scalability of MOEA/DD for large scale problems. Algorithm 2.7 illustrates the MOEA/DD algorithm.

Output: population *P*
01: [*P,W, E*] ←INITIALIZATION() ;
// *P* is the parent population, *W* is the weight vector set and *E* is the neighborhood index set
02: while *termination criterion is not fulfilled* **do**
03: for $i \leftarrow 1$ **to** *N* **do**
04: *P* ←MATING_SELECTION*(E(i), P)*;
05: *S* ←VARIATION*(* $\overline{P}$ *)*;
06: **foreach** $\mathbf{x}^c \in S$ **do** // $\mathbf{x}^c$ is an offspring
07: *P* ←UPDATE_POPULATION*(P,* $\mathbf{x}^c$*)*
08: **end**
09: end
10: end
11: return P

Algrithm 2.7 The MOEA/DD algorithm (Li et al. 2015)

2.4.3.4 Two_Arch2 algorithm

Two_Arch2 (Wang et al. 2015) is a low-complexity algorithm that represents an enhancement of the Two_Arch algorithm that offers two significantly improved archives (CA and DA) focused on diversity and convergence and separately designed to provide a more balanced optimization algorithm for ManyOPs. In Two_Arch2, different selection principles based on Pareto based indicators are attached to both archives. Moreover, a new diversity maintenance scheme based on the Lp ($p < 1$) standard for ManyOPs is proposed. Illustrated in Figure 2.9, the Two_Arch2 is based on indicator and Pareto dominance, which combines the benefits of the indicator and Pareto-based MOEA. Indeed, the $I\varepsilon+$ indicator is used as a selection principle in the CA archive to improve the convergence of MaOPs while the Pareto dominance is used as a selection principle in DA to promote diversity. This means that Two_Arch2 has a good convergence capability by CA (that is indicator-based), and it can maintain a satisfactory diversity by DA (that is Pareto-based). In Two_Arch2, an $L_{1/m}$-norm maintenance scheme is used to remove additional DA solutions since most Euclidean distance-based diversity maintenance methods encounter problems in solving ManyOPs. Hence, the Lp-normbased distances ($p \geq 1$) are inefficient in a high-dimensional space whereas the distances based on the norm Lp ($p<1$) are more efficient in such high-dimensional space.

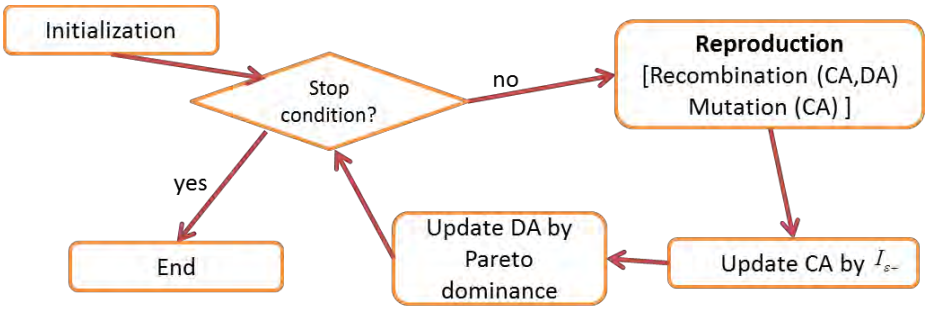


Figure 2.9 Flow-chart of the Two_Arch2 Algorithm

According to the authors of Two_Arch2, its experimental results show an efficiency for ManyOP up to 20 objectives with diversity, convergence and satisfactory complexity. Despite this, unsatisfactory performance of Two_Arch2 on several issues such as the WFG. Moreover, unlike other recent optimizers such as NSGA-III, the Two_Arch2 could not keep the extreme points. In this respect, the extreme point maintenance principle in NSGA-III can be used to enhance the Two_Arch2.

2.4.3.5 Similitudes and differences between many-objective algorithms

Different similarities and differences between MOEA/DD, MOEA/D and NSGA-III can be detailed as follows (Li et al. 2015):

With regard to MOEA/DD and MOEA/D, the main similarities concern the use of a set of weight vectors to guide the selection procedure, the use of the concept of neighborhood and the application of a scalarization function to measure the aptitude value of a solution. The main differences between MOEA/DD and MOEA/D are as follows:
- MOEA/DD use weight vectors which, in addition to defining a sub-problem that evaluates the aptitude value of a solution, specifies a subregion for estimating the local density of a population.
- The MOEA/D uses an update/selection procedure with a 'one-sided' selection in which only the sub-problems have the right to choose their preferred solutions. On the other hand, each MOEA/DD solution is associated with a subregion (subproblem) and has the right to select its preferred subregion.
- The speed of convergence of the MOEA/D is often faster than that of the MOEA/DD. However, MOEA/D does not enhance the preservation of diversity, which is very important for many-objective problems.
- In MOEA/D, the concept of neighborhood is used for mating and updating while it is used only for mating restriction in MOEA/DD.

With regard to MOEA/DD and NSGA-III, the main similarities concern the selection procedure where the two algorithms use the same concept (named weight vectors in MOEA/DD and reference points in NSGA-III) with solutions that are associated with weight vectors (or reference point). Moreover, the two algorithms use the Pareto dominance relation to divide the population into several levels of non-denomination. The main differences between MOEA/DD and NSGA-III are as follows:
- The selection in NSGA-III is based on a generational scheme while the selection in MOEA/DD is based on a permanent scheme.
- In MOEA / DD, the population is divided into several levels of non-denomination with a selection procedure that does not fully obey the decision taken by the Pareto dominance relationship and a solution associated with an isolated subregion may survive the next iteration even if it does not belong to the last level of non-denomination. On the other hand,

such a solution in NSGA-III cannot survive the next generation and can result in a significant loss of population diversity.

2.4.3.6 Difficulties in MaOAs

Recently, with an increasing number of objectives, the realism and complexity of optimization problems, the interest of the EMO (evolutionary multi-objective optimization) community are focused on the evolutionary many objective optimization (EMaO). This focus is explained by the fact that the performance of most evolutionary optimization algorithms deteriorates if the number of objectives exceeds three.

In what follows, we enumerate the most critical challenges faced by EMO and EMaO (evolutionary manyobjective optimization) when solving MaOPs:

***Exponential complexity in space and time:** A major failing of most EMOs is the time exponential increase with the rise in the number of objectives. Same goes for the exact calculation of certain performance indicators such as the Hypervolume (HV) (Zitzler et al. 1999). This exponential complexity can deteriorate the performance of EMOs in the resolution of MaOPs. As it is the case in our study, this problem can be solved by using dimensionality reduction mechanisms to eliminate redundant objectives. An approximate method is also used to calculate the performance indicators.

*** Pareto-based EMOs:** The final solutions given by Pareto-based EMOs for many-objective problems can be distributed uniformly in the search space and away from the desired PF (Pareto Front). Studies like (Purshouse et al. 2007) show that random search can yield better results than Pareto-based EMOs for problems having nine objectives or more. Hence, the criterion of dominance by Pareto cannot be applied to compare individuals. In this respect, several solutions can be used to solve this problem. We can cite, for instance, replacing the principle of Pareto dominance by that of the $\in$-dominance (Laumanns et al. 2002). Moreover, modifying the dominance region by the control of the dominance angle to increase the selection pressure in the PF. Another solution used to guide the population towards the PF is the maintenance of diversity. As in the case of our work, this can be achieved by incorporating some mechanisms to ensure that the parents are chosen in the neighborhood when performing the mutation and recombination operators or by adaptively selecting these operators.

*** The problem of representing the trade-off surface:** A problem with n objectives has a PF with (n-1) hypersurface dimensions, which considerably increases the number of points used in the representation of this surface and creates problems related to the visualization of these points. Hence, new visualization methods of the PF are needed.

*** The recombination and mutation operators are inefficient:** By increasing the number of dimensions in the search space, the distances between individuals become large, causing the inefficiency of the operators (Purshouse et al. 2007). For this reason, having distant parents implies that the created offspring may also be distant from its parents, which slows down the search process. To solve this problem, we adaptively choose the operators at each iteration according to their contributions.

*** The density estimation is inaccurate:** If the space is highly dimensional, the degree of crowding of the population will become unspecifiable by the majority of the density estimation methods. For example, the density estimation using the crowding distance loses its relevance for a number of objectives greater than three which can be explained by the exponential increase in the number of hyperboxes in the grid that causes the dispersion of individuals in several hyperboxes. Indeed, several MoAs suffer from the inaccuracy of the density estimators using the euclidean distance to measure the resemblance between individuals. Thus, it is necessary to substitute the dominance relation by other concepts such

as the ∈-dominance (in ∈-NSGA-II (Kollat et al. 2005)) or the reference points (in NSGA-III (Deb and Jain 2014))

2.4.4 Hybridization of Meta-heuristics

In order to improve the performance of an algorithm or to fill some of its inconvenient, it may be combined with one or more method. In fact, these cooperative approaches allow the optimization methods combining their benefits to improve their performance in order to obtain better results. Hybrid approaches can be divided into two groups: The hybridization of metaheuristics and the hybridization of exact methods and metaheuristics. In what follows, we present the main hybridization schemes.

2.4.4.1 Hybridization between metaheuristics

This hybrid approach consists in combining several different metaheuristics. Local search, annealing simulated, taboo research and evolutionary algorithms have already been successfully hybridized in several applications. In (Talbi, 1999), the author proposed a classification taxonomy of these different hybridizations. The idea of classification is based on a qualitative comparison of hybrid metaheuristics. The taxonomy has two aspects: A hierarchical classification which allows identifying the structure of hybridization. Then, a general classification which specifies the details of the algorithms involved in hybridization.

* **Hierarchical classification**: This classification is based on two classes: low-level hybridization and high-level hybridization. A low-level hybridization is obtained when a function of a metaheuristic is replaced by another metaheuristic. A high level hybridization is achieved when two metaheuristics are hybridized without relation between their internal functioning. Each of the two previous hybridization classes is subdivided into two other classes: relay hybridization and coevolutionary one. The relay hybridization occurs when metaheuristics are executed sequentially, i.e. a metaheuristic uses the result of the previous one as input. Coevolutionary hybridization occurs when agents cooperate in parallel to explore the solution space. Subsequently, as shown in Figure 2.10, there are four different classes of cooperation of all previously named classes: low-level relay, low-level co-evolutionary, high-level relay and high-level co-evolutionary.

First of all, low-level relay hybridization represents algorithms in which a metaheuristic is incorporated into another single-solution metaheuristic. Then, the low-level co-evolutionary hybridization consists in incorporating an exploitation-based local search algorithm into an exploration-oriented population metaheuristic. Afterward, high-level relay hybridization is achieved when complete metaheuristics are performed sequentially. Finally, high-level coevolutionary hybridization involves a set of metaheuristics that work in parallel and cooperate to find the optimal solution to a problem.

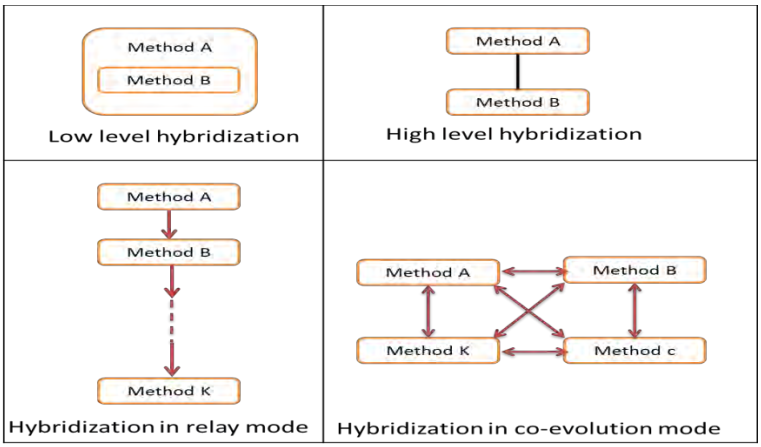


Figure 2.10 Classes of Hybridization

*** General classification:** According to (Talbi, 2000), there is widespread classification taxonomy for hybrid metaheuristics with three dichotomies: homogenous vs. heterogeneous, global vs. partial and specialized vs. general hybrid approaches. Hybridization is said to be homogeneous when the combined metaheuristics are identical. Conversely, heterogeneous hybridization combines different metaheuristics. Global hybridization ensures that all metaheuristics explore the entire solution space while partial hybridization decomposes a problem into sub-problems having their specific solution space. Then, each sub-problem is given to an algorithm. Finally, the general hybridizations are those where all the algorithms solve the same optimization problem. On the other hand, the 'specialized' hybridizations are those where each algorithm solves a different optimization problem.

2.4.4.2 Hybridization of Meta-heuristics and Exact Methods

Exact methods seek to find the optimal solution by explicitly or implicitly examining the entire search space. Typically, exact methods are dedicated to solving small problems, while metaheuristics adress a wide range of multi-objective problems. However, hybrid metaheuristic-exact approaches can combine the advantages of these methods to improve their performance and provide optimal solutions. There are several ways of hybridization. (Puchinger and Raidl, 2005) proposed a classification of metaheuristic-exact cooperation that is divided into two categories: collaborative and integrative hybridizations.

*** Collaborative hybridization**: Using this type of hybridization, the algorithms exchange information sequentially, in parallel or in an interlaced manner. Indeed, the sequential execution is done in such a way that the exact method is a pretreatment of the metaheuristic, or vice versa. For the interlaced or parallel performance, the two approaches are independent and deal with different parts of the problem.

*** Integrative hybridization**: The integrative combination of an exact method and a metaheuristic is done in such a way that one of the two algorithms is an integrated component in the other one. Therefore, we can integrate an exact method with a metaheuristic and vice versa.

2.4.5 Relationship between the deployment problem and the multi-objective optimization

In this section we justify the choice of using meta-huristics and evolutionnary optimization algorithms as an approach to solve the problem of 3D indoor deployment of nodes. Indeed, the deployment is a real-world complex problem that involves several often antagonistic objectives and different constraints (of localization or routing for example).

The deployment problem is proven to be NP-Hard *Cheng et al. (2008)*. As a result, there is no exact algorithm that can solve it in a polynomial time if the size of the problem exeeds a certain threshold. Most exact algorithms such as branch and bound provide an effective solution for the deployment problem only if it is a small problem (if the number of nodes does not exceed three).

Moreover, compared with geometric approaches (Voronoi Partitions, Virtual Forces) resolving the deployment, the optimization using meta-heuristics allows the user (DM) to choose between multitudes of possible solutions and even to guide the search towards the desired solution/solution set.

In addition, the evolutionary optimization approach is suitable for the deployment problem due to the distributed, dynamic and stochastic nature of this problem; and the existence of a research space containing a large number of possible solutions that are close to each other. The challenge is therefore to choose the closest to optimal solution.

When solving the problem of 3D deployment, aside from the exponential increase in cost (in terms of time and space), even Pareto-based EMOs become inefficient, the same for the recombination operation. Indeed, it was even proven (Purshouse et al., 2007) that, if we

increase the size of the problem (by increasing the number of nodes and the RoI to cover in our case), the performence of the pareto-based algorithms degrades until the random choice of node positions becomes more efficient than these latter algorithms. For more details concerning the difficulties encountered by the multi-objective optimization algorithms when solving complex real-world problems, refer to the section 2.4.3.6. In this regard, to enhance the performance of the optimization algorithms to solve the 3D indoor deployment, a set of modifications and justified hybridization schemes between these algorithms are proposed as contributions in this dissertation.

2.5 Multi-agent Systems

In recent decades, multi-agent systems (MAS) have become increasingly important in computing as well as in artificial intelligence. Thanks to their distributed architecture, MAS become more responsive and have been used in different areas such as information retrieval, robotics or simulation of artificial life. Thus, MAS facilitate the construction of a distributed, heterogeneous, flexible architecture, and they are able to offer a high quality of service in a collective workspace. The fundamental issue of MAS is the theoretical and experimental analysis of mechanisms of self-organization when multiple autonomous entities interact and to the realization of artifacts able of accomplishing complex tasks by cooperation and interaction. A multi-agent system (MAS) is a decentralized system composed of a set of autonomous agents, located in a specific environment to achieve a global goal. MAS derived from the distribuded *artificial intelligence* (Carabelea et al., 2004).

2.5.1 Notion of Distributed Artificial Intelligence

Distributed Artificial Intelligence (DAI) is a recent variant of artificial intelligence that adds the dimension of interaction and distribution of software components instead of having a single centralized software. The DAI goes from modeling a single intelligent agent to the modeling of a set of agents, generally in a cooperative relationship, to achieve a specific objective. This objective concerns usually a problem to be solved that is distributed and divided into a set of subproblems.

2.5.2 Concept of agent

There is no standard definition of an agent. As shown in Figure 2.11, an agent is a physical or virtual entity capable of acting in a specific environment to which it was designed. A physical entity like a robot, car, plane, thing (in IoT) and a virtual entity like a software, a computer module. An agent can communicate directly with another agent through messages or with stimuli. According to (Carabelea et al., 2004), an agent is defined as follows:

-An entity that perceives its environment and acts on it.

- A computer system, located in an environment, autonomously acting in order to achieve the goals (purpose) for which it was designed.
- An entity that operates continuously in an environment where other processes take place and other agents exist.

An agent has the following characteristics:

- Able to act in an environment.
- Can communicate directly with other agents.
- Has its resources, has skills and offers services.
- Able to perceive (but in a limited way) his environment and has a partial representation of this environment.
- Can reproduce.
- Its behavior tends to meet its objectives, taking into account the resources and skills it has, and according to its perception, its representations and the communications that it receives.

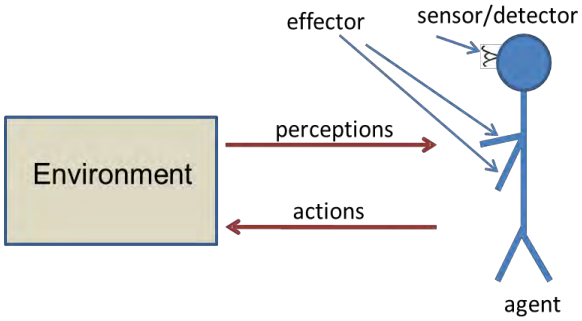


Figure 2.11 Schematization of an agent

2.5.3 Multi-agents systems

MAS are systems grouping more than two agents and characterized by the following properties (Xie and Liu, 2017):

- An environment E: each agent can perceive his environment and cannot act in another space.
- A set of object O which defines the agent's entourage.
- A set of object A which forms the set of agents with which an agent can communicate and interact.
- A set of relations R which define the existing relations between agents.
- A set of OP operation enabling the agents to move, communicate, consume, etc.

Hence, MAS (Figure 2.12) are based on the following principle: MAS = {Agent + Environment + Interaction + organization}.

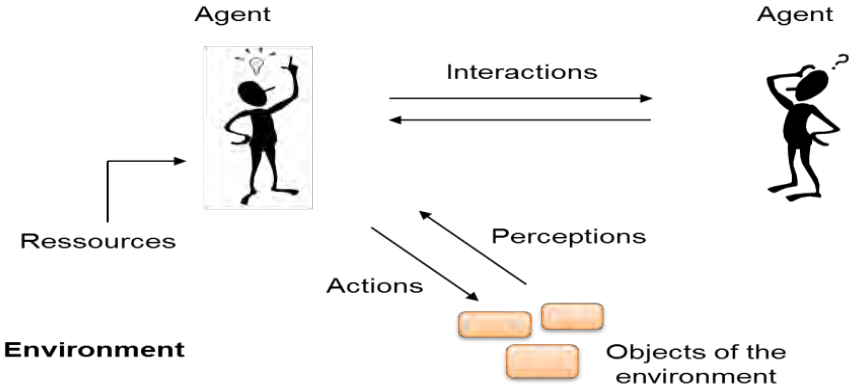


Figure 2.12 Multi-agent System

*** MAS characteristics:** Several characteristics are related to the agent concept. The most interesting features are as follows:

- Autonomy: this is one of the leading characteristics of MAS. It implies that the elements of MAS have their autonomy. Indeed, agents are not guided either by commands or by the human intervention.
- The capacity to act: an agent is conceived for a specific treatment. Hence, it is guided by a set of objectives. It must not be influenced by the interactions of its environment.
- Reactivity: a reactive agent is used to achieve a task for which it receives orders or stimuli from its environment.
- Perception: the agent may have a vision of its local environment and the agents in this environment.
- Sociability: depending on the nature of the environment, the agent interacts with other agents or human beings to accomplish a task.
- Anticipation: the agent can perceive, in a limited way, future events.

- Adaptability: the agent must be adapted to any changes in its environment.
- Communication: the communication between agents is necessary to define when conceiving MAS. In a cognitive system, communication is ensured through messages whereas in a reactive system, the communication is achieved by the diffusion of signals.
- Learning: it allows to agents to acquire new knowledge after the communication phase.
- Cooperation: When cooperating, agents are engaged in a joint action aiming to facilitate and accelerate the execution of commun task.
- Interaction: is a primary feature of MAS which serves to define a dynamic relationship between two or several agents. This characteristic gathers the collaboration and the coordination.
- Distribution of tasks: considering the limited capacities of an agent in energetic, computation, and memory resources, and the complexity of problems, agents are forced to divide the problem into sub-problems.
- Decentralization: It makes the MAS more reliable and robust. It can overcome the failure of a single agent.

*** Advantages and disadvantages of MAS:** The use of MAS has a set of advantages such as the distributed solving of problems and the possibility of obtaining a stable solution following the decomposition of the problem into independent sub-problems and the collection of these partial solutions. Besides, MAS can resolve individual failures of agents without much affecting the performance of the overall solution. These systems also give the possibility of practically realizing situations and experiences which are difficult to achieve in reality. However, MAS has some limitations such as the complexity, the difficulty of modeling, issues related to the gathering of solutions and the implementation.

2.5.4 Agents types

Agent differs in the functionality for which they are designed. Agents can be classified as follows:
- **Cognitive agents**: an agent is considered as intelligent, has a knowledge base containing the necessary information for its operation. It has the capacity of reasoning and communicating. Agents can simulate human behavior.
- **Reactive agents**: a reactive agent is guided by a set of rules and reacted only to respond to a query or a stimulus. It does not have any means of memorization.
- **Mobile Agents**: a mobile agent is an agent that moves in its environment to do a treatment or giving information.
- **Collaborative agent**s: these are autonomous agents who work together to solve a problem.
- **Internet agents**: this type of agents is more used mainly in the searching, gathering and collecting data, or administering websites.
- **Interface agents**: this type of agents assists the user and can be adapted to its habits.
- **Hybrid agents**: represent agents that combine more than two characteristics such as mobility, responsiveness and collaboration.

2.5.5 Software platforms for MAS

A platform is a tool that allows the development of agents. Actually, many platforms exist; the most popular are the following ones:
- **JADE**: it is an open source platform developed in java. Created by Gruppo Telecom laboratory in order to realize cognitive MAS according to the FIPA standard. The latter standard establishes the normative rules that allow to a company of agents to interoperate. Agents communicate with each other using the ACL language.
- **MADKIT**: is an evolutionary platform used to develop MAS, especially MAS based on organizational criteria. It is based on the AGR model (agent/group/role).

- **Jack**: is an environment for the development, execution and integration of commercial MAS. Developed in Java, Jack is characterized by its strong focus on programming agents.
- **Jadex**: a new platform for building goal-oriented agents according to the BDI model (belief-desire-intention).

2.5.6 Applications of MAS in WSN

WSN can be used to collect information from environments to which the human being does not always have access. A WSN is a system consisting of several nodes that may acquire data of its environment, to communicate with each other, to achieve a specific number treatment. Most of these features are better resolved with MAS. The use of MAS can solve real-world problems such as routing and energy consumption.
- **Sensor deployment**: (Mekni and Haddad 2010) adressed the problem of sensor web deployment, considered as a spatial problem because of the high-influence of the geographic specificities of the space on the nodes. As an approach, a geo-simulation framework based on a multi-agent system is proposed to simulate sensor deployment in Informed-Virtual-Geographic Environments. Moreover, Qi et al. (2015) aim to resolve the problem of modelling and designing large-scale multi-agent systems deployed in 3D spaces. The proposed method relies on partial differential equations (PDEs) controle.
- **Routing**: The routing problem consists in establishing an optimal routing of packets across the network in the sense of a particular performance criterion such as energy consumption. Although, sometimes we do not seek a minimal route but a distributed loads between nodes. MAS composed of a large number of nodes are characterized by an unstable topology. Thus, MAS needs to be adapted to any change in the network and must establish the shortest routes between different nodes. Several routing protocols have been developed.
- **Clustering organization**: when the size of the network is important, its management becomes difficult. Clustering is often used to optimize the energy consumption and the routing. A cluster is defined by a set of nodes according to several criteria such as the geographical position and the connection between nodes.

2.6 Preferences incorporation

2.6.1 Generalities

In general, the DM needs only a small number of non-dominated solutions, especially for real-world problems. On the other hand, the result of a MaOA is a set of non-dominated uniformly distributed solutions. Moreover, since the user does not need to know all the points on the PF and knowing that the population size and the number of solutions necessary to approximate the hole PF increase exponentially with the number of objectives (Ishibuchi et al. 2008), it is interesting to focus the search on some specific regions according to the user's preferences. Besides, in real-world problems, the size of the population is usually too limited compared to the high-objective space which prevents producing meaningful results (Rachmawati et al. 2006). Hence, incorporating preferences aim at resolving the problem of low selection pressure for convergence by carrying out a preference ordering over non-dominated solutions. In fact, most of the studies presented in literature either focus on only a part of the PF, which improves convergence but decreases diversity, or has a high computational time which increases exponentially as the number of objectives rises. A review that discusses the studies investigating the preference approaches is available in (Ishibuchi et al. 2008).
When designing a preference-based method, two issues must be considered: when integrating the preference information? And what is the model of preference? Different preference models can be cited, such as the *objective rankings*, the *trade-off between objectives* and the *goal specification* (Rachmawati et al. 2006).

Based on the time of integrating the preference information, preference methods can be classified into three main categories (Jaimes et al. 2011):
- **A priori algorithms** where the preference information which directs the population to converge to a PF subset is known before starting the search for solutions.
- **Interactive algorithms** where the process of optimization asks the DM to introduce the preference information interactively to guide the search to a specific PF subset. In fact, only a few studies aiming at progressively incorporating the user's preferences following the evolving of the new solutions were proposed.
- **A posteriori algorithms** where the preference information is given after executing the MaOEA.

Because they focus the search orientation on the region of interest, the interactive and a priori algorithms are more likely to pay more attention to preferred solutions and reduce the computational cost during the search process. A posteriori preference-based approaches might give a large number of solutions which are non-interesting to the DM. In our works, as an interactive preference reduction method, we use PI-EMO-PC (Sinha et al. 2014). Despites their numerous advantages, the major weakness of the interactive techniques is that the algorithm used in such techniques need to interact frequently with the DM who can become tired, which leads, in some cases, to misleading information about the preference information provided by the exhausted user (Gong et al. 2013). Thus, to overcome this limitation in our work, we suggest a hybrid preference process.
The progressive engagement of the preferences (interactive algorithms) is more efficient since it allows the DM to adjust his preferences during the intermediate generations of an algorithm (Sinha et al. 2013). Therefore, a model based on an enhanced PI-EMO-PC is used as a 'preference method' in our work.

2.6.2 Interactive preferences (PI-EMO-PC)

As an instance of interactive preferences methods, we can mention the PI-EMO-VF (Deb et al. 2010) based on an approximate value function where the EMO algorithm dispose of a set of well-sparse non-dominated solutions progressively generated after every few generations and ask the DM to provide his preference information about the relation of one solution over another. Ideally, the DM gives a complete ranking of solutions (from best to worst). Otherwise, he provides partial preference information. The given preference information is employed to construct an increasing polynomial value function. According to the expected progress and the constructed value function, a termination condition is set up. In fact, the proposed PI-EMO-VF is a generic procedure that can be incorporated into any existing EMO. However, as mentioned by its authors, a set of parameters is expected to be specified by the user such as $\boldsymbol{\tau}$ (the number of EMO iterations after which the PI-EMO-VF is called), $\boldsymbol{\eta}$ (the number of desired non-dominated solutions for which a preference information is given) and $\boldsymbol{d_s}$ (the expected improvement in the solutions from the current best solution based on the value function. From the aforementioned ideas, we may conclude that PI-EMO-VF method has several drawbacks as it: ***i)*** needs a partial or complete ordering of the set of points, ***ii)*** requires to solve another optimization problem in order to construct the DM's value function and to know (as a parameter in the algorithm) the number of generations after which the DM call will be performed. Moreover, the unknown number of DM calls at the beginning of the algorithm.
To design our proposed hybrid preference algorithm, as an interactive preference, we use another recent variant of the PI-EMO, called progressively interactive EMO based on polyhedral cones (PI- EMO-PC). Indeed, like the PI- EMO-VF procedure, PI-EMO-PC was implemented on the NSGA-II algorithm. But, it is possible to integrate it with any other multi-objective EMO algorithm. Besides, PI- EMO-PC can successfully handle the PI-EMO-VF

drawbacks by simultaneously reducing the number of needed parameters in PI-EMO-VF and providing better performance. PI-EMO-PC replaces the process of constructing a value function by using polyhedral cones to modify the domination criteria of an EMO and guides it towards the region of interest which is a single most preferred point on the high-dimensional PF. PI-EMO-PC needs as input a parameter T_{DM}, representing the maximum number of times the preference information is required in order to be provided by the DM, and a parameter $|A|^{max}$ designating the maximum archive size. The best member of the archive set is chosen according to the preference information. The archive A is initially empty. Then, after each generation, it will include all the found feasible non-dominated solutions. If the size of the archive exceeds its maximum, a k-mean clustering will be applied to keep the diverse set of $|A|^{max}$ clusters and the rest of the solutions will be deleted. Despite the fact that the PI-EMO-PC procedure was developed using the NSGA-II algorithm, it is a generic procedure which can be integrated into any other multi-objective EMO algorithm. However, the PI-EMO-PC suffers from some drawbacks such as the need to know the number of DM calls in advance and the risk to have wrong directives if the DM becomes tired. To solve these problems, we introduce a hybrid preference procedure which requires fewer parameters and more flexibility when interacting with the DM.

2.7 Dimensionality reduction

2.7.1 Generalities

As an example of approaches used to overcome the complexity of many-objective problems, we can mention the dimensionality reduction which assumes the existence of a set of redundant objectives in a given M-objective optimization problem. Based on the objective vectors of the obtained non-dominated solutions, the dimensionality reduction approaches aim at identifying the smallest set of conflicting objectives which is smaller than the original set of objectives and that generates the same PF as the original problem. Since the same PF is preserved, the reduction does not cause information loss (Saxena et al. 2013). In these approaches, the resulting objectives are considered as necessary, while others are regarded redundant. Relying on the study of the correlation structure between the objectives and the dominance relations, the dimensionality reduction approaches allow reducing the number of objectives to three or less. Hence, an unsolvable problem will become solvable with a lower computational cost, higher search efficiency and easier visualization. Therefore, if a high-dimensional many-objective problem and a lower-dimensional one have similar PF, then the latter can be optimized instead of the former one.

Considering the time of incorporating it into the MaOEA, a dimensionality reduction method can be classified either as an offline or online method. In offline methods, dimensionality reduction is performed after obtaining Pareto optimal solutions. Offline methods are classified into three sub-categories (Lan et al. 2010): feature selection-based, dominance structure-based and correlation-based methods. However, in online methods, dimensionality reduction is carried out gradually. In other words, in each iteration, a set of solutions is obtained and the dimensionality reduction method is invoked. Applying such methods, the number of objectives can be reduced progressively during the search process.

The current reduction methods can be summarized as follows:

- Machine Learning-based (Deb et al. 2006): consists in using machine learning techniques, such as Maximum Variance Unfolding (MVU) and Principal Component Analysis (PCA), to eliminate the dependencies of the second and higher order in the non-dominated solutions respectively.

- Dominance Relation Preservation-based (DRP): As introduced in (Brockhoff et al. 2006), this reduction method relies on the preservation of the dominance relations in the given non-dominated solutions.
- Pareto Corner Search based (PCS): Singh et al. (2011) proposed a Pareto corner search evolutionary algorithm (PCSEA) which aims at finding only the corners of the PF instead of searching the complete PF assuming to adequately investigate the dependencies of the different objectives.
- Unsupervised Feature Selection-based (UFS): This method proposed in (Jaimes et al. 2008) examines the correlation among objectives by considering the distance between objectives as a metric to evaluate the conflict between them.

The dimensionality reduction methods are efficient in the sense that they: ***i)*** can be easily and consistently be combined with other approaches, ***ii)*** help the DM understand the many–objective problem by eliminating redundant objectives and ***iii)*** minimize the computational cost of the many–objective evolutionary algorithm.

However, dimensionality *reduction* methods suppose the existence of redundant objectives, which may restrict the applicability of these approaches when the problem to be solved has no redundant objectives. Consequently, dimensionality reduction techniques may fail to minimize the number of objectives or give a set of solutions which does not cover the complete Pareto front. In our works, as a reduction method, we use machine learning techniques MVU-PCA (Principal Component Analysis and Maximum Variance Unfolding) which is an offline correlation-based reduction method.

2.7.2 Offline correlation reduction methods based on machine learning for the 3D deployment problem

Machine learning-based objective method is a reduction approach relying on the fact that the structure of a high-dimensional data may be transformed in order to minimize the effect of noise (non-optimal solutions that may be different from those which define the true POF) and dependencies (redundancy) between the different objectives.

Our studied problem, the 3D Deployment of WSNs, can be viewed as a machine learning objective reduction problem due to:

- The redundancy and the presence of non-conflicting and correlated objectives.
- The structure of the PF of our problem which refers to its intrinsic dimensionality (m) and essential components.
- The garbled high-dimensional data which refers to the non-dominated solutions resulting from the EMO algorithm, generally providing a poor POF-approximation. Thus, correlated objectives on the POF can show a partial conflict in the proposed EMO solutions.

2.8 Conclusion

This chapter has been devoted to describe the different types of optimization problems and their classifications. A presentation of the various resolution approaches existing in the literature, such as; the aggregation approach, the ε-constraints approach and the goal-based programming approach is given. Indeed, when the number of objectives is higher or the problem is large, meta-heuristics; despite giving approximate solutions; are less demanding in terms of computation time and execution time. Thus, they are the most used for the resolution of real problems. Moreover, we detailed the characteristics of MAS, as well as the dimensionality reduction. And finally, we gave an idea about the incorporation of the user preferences. The next chapter is dedicated to present the proposed mathematical formulation of the 3D indoor deployment problem relying on real assumptions and constraints taken from our real prototyping tests. Then we present the proposed modifications and hybridization schemes aiming at improving the performance of the tested algorithms in the context of many-objective complex real world problems.

Part II
Contributions

Chapter 3
Theoretical contributions

3.1 Introduction

To resolve our problem, we suggest an Integer linear programming formulation. Indeed, a set of constraints, like the distribution of nodes, the number of hops between them, the distances and paths and the quality of links, were considered. Different parameters, such as the initial deployed node locations, the cost of deploying a node, the transmission signal strength and the minimal required power of the received signal to detect a target, were taken into account in our model. Moreover, we illustrate in this chapter the specifications of the proposed modified algorithms. Firstly, we detail the manner of coding the chromosomes used in the evolutionary considered algorithms. Afterwards, we present the proposed modifications incorporated in the many-objective algorithms. Among these modifications, the use of the neighbourhood and an adaptive guided concept for the mutation and recombination operators, and the use of multiple scalarizing functions in the aggregation based approaches. Moreover, we present the different hybridization schemes aiming at improving the behavior of the tested algorithms and enhancing their performances in solving complex real world problems. These hybridization include, among others, the incorporation of the reduction of dimentionality, the users preferences and multi-agent systems.

3.2 Mathematical model of the 3D indoor deployment in DL-IoT collection networks

3.2.1 Architecture of nodes, assumptions, notation and objective function

3.2.1.1 Architecture of nodes

We consider the following types of nodes:
- Stationary nodes composed of the set of fixed nodes initially installed. This type of nodes can be randomly disseminated. But, it is better to adopt a strategy to distribute them according to the applicative objectives.
- Nomad nodes added to enhance the 3D deployment scheme. Their locations are determined by the proposed genetic algorithms.
- Mobile nodes (targets) include a set of persons to control. They are equipped with a sensor transmitting and receiving signals.

3.2.1.2 Assumptions

The following assumptions are considered:
- An ***anchor node*** is defined as a node having a known position. It can be an initial sensor placed in a specific location, a nomad sensor or a sensor attached to a mobile target. This assumption was imposed by the used localization protocol.
- Conforming to the used 3D localization protocol, we supposed that four anchors were needed to locate each mobile target. Indeed, we propose a hybrid localization scheme that improves the used range-free technique (3D DV-Hop) by introducing a range-based localization (RSSI data). First, the 3D DV-Hop used the information of the network connectivity to estimate the 3D positions of the nodes. Then, the RSSI value was applied to correct both these positions found by the 3D DV-Hop according to the signal strength and the distances between each mobile node to be localized and its four nearest anchors. As it will be mentioned later in the experimentations, we measured the RSSI value in two directions: i) that received from other fixed or nomad nodes was measured starting from each mobile node, and ii) that received from the concerned mobile node was measured beginning from each fixed or nomad node. The final admitted RSSI value between the mobile node and each other node is the highest. Figure 3.1 shows the proposed localization principle using the 3D DV-Hop and the RSSI algorithm.

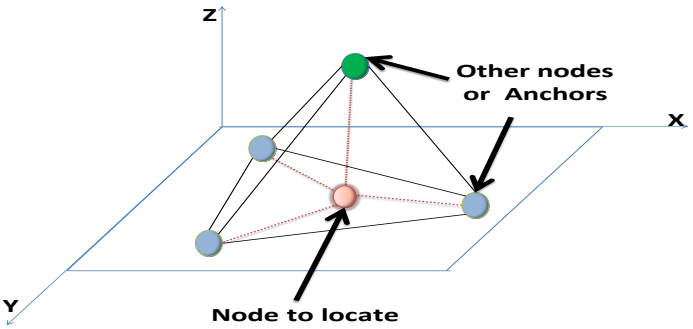


Figure 3.1 The 3D DV-HOP and RSSI protocol

- Rooms and halls are supposed to have heterogeneous 3D spaces.

3.2.1.3 Notation

The following sets, decision variables and parameters were used:

- **Sets**

S: is the set of potential sites where sensor nodes can be installed, $\mathbf{S} = \mathbf{S_a} \cup \mathbf{S_b}$ such that "$\mathbf{S_a}$" is the set of potential sites where stationary sensor nodes can be installed. "$\mathbf{S_b}$" is the set of potential sites where the nomad sensor nodes can be installed. Note that a site must not be in both sets. Thus, $\mathbf{S_a} \cap \mathbf{S_b} = \varnothing$.

N: is a set that denotes the different types of the nodes. Let $\mathbf{N} = \mathbf{N_a} \cup \mathbf{N_b}$ such that "$\mathbf{N_a}$" is the set of different types of stationary nodes and "$\mathbf{N_b}$" is the set of different types of nomad nodes. We can use various types of sensors having different functionalities which can be gathered in the same sensor like detecting the degree of temperature, the degree of luminosity or the opening and closing of doors.

T: is the set of mobile targets to be detected; "$\mathbf{t_k}$" is a target.

V: is the set of nodes having different types in N and deployed in several sites in S.

K is the set of scheduling periods when a sensor i∈**V** is activated.

- **Decision Variables**

$Sg_{ss'}$ equal to 1 if the node positioned at a site s ∈ S receives a signal from another node positioned at a site s' ∈ S with a power of transmission greater than or equal to the minimum required power to detect it.

$T_{ss'}$ equal to 1 if the node positioned at a site s ∈ S transmits a signal from another node positioned at a site s' ∈ S with a power of transmission greater than or equal to the minimum required power to detect it.

X_{ts} equal to 1 if a node positioned at a site s ∈ S can receive a signal from a target located at a position t ∈ T with a power of transmission greater than or equal to the minimum power required to detect it.

$SP^{ss'}$ represents the shortest path (in terms of hops) between two nodes situated in two sites s ∈ S and s' ∈ S.

$L_{ss'}^{ab}$ equal to 1 if the link (s, s') is used to route the traffic flow from source node a ∈ S to a destination node b ∈ S, knowing that $SP^{ss'}$ is the shortest path (in terms of hops) between a and b.

Pos_{ijk} a real variable representing the 3D coordinates (i,j,k) providing the potential position of a sensor in the indoor space.

$CovP_{ijk}$ equal to 1 if and only if the position Pos_{ijk} is covered by a node with a power of transmission greater than or equal to the minimum required power to detect it.

Pfx_s^n set to 1 if a fixed sensor having a type n ∈ N is positioned at a site s ∈ S; 0 otherwise.

Pnd_s^n set to 1 if a nomad sensor having a type n ∈ N is positioned at a site s ∈ S; 0 otherwise.

Pmb_s^n set to 1 if a mobile sensor having a type n ∈ N is positioned at a site s ∈ S; 0 otherwise.

- **Parameters**

r corresponds to the radius of a sensor.

nbT denotes the number of mobile targets. "nbF" , is a parameter representing the number of initially-deployed stationary anchors. This number can be set by default to a random number or to $n_{\min} \times (nm / 2\Pi r^2)$. "nbN" is a parameter representing the number of nomad nodes needed to be added. Let N_{max} be the maximum number of nodes that can be deployed within the wireless network. Thus (nbT+nbF+nbN) ≤ N_{max}.

C_s^n is the hardware cost of a node (including price) having a type n ∈ N and installed at a site s ∈ S.

E₀ is the initial remaining energy of the sensors. We assume that sensors have the same E_0. Bt_i is the remaining energy in the battery of the sensor i at an instant t.

n_{min} stands for the degree of coverage. It defines the minimum number of nodes receiving a signal from a target to localize it. When using the proposed hybrid 3D localization model (based on 3D DV-Hop and RSSI), the parameter n_{min} is generally set to 4.

$\delta_{ss'}$ represents the attenuation ratio between two nodes in two sites s ∈ S and s' ∈ S.

γ_{ts} designates the attenuation ratio of the signal from a target t ∈ T to a site s ∈ S.

P_t represents the power of transmission of a target having a position t ∈ T (in watts).

$P_{\min}^n$ corresponds to the threshold for the receiver power. It is the minimum required power of the signal (RSSI) transmitted by (received from) a node having a type n ∈ N to detect it.

$hp_{\max}$ is a parameter defining the maximum allowed number of hops between any nomad node and a stationary node installed at site s ∈ S_a.

M_s^m stands for the required number of hops between the nomad node 'm' and the nearest stationary node installed at the site s ∈ S_a.

Q_s^m designates the number of nodes (stationary or nomad) installed at site s ∈ S for which a nomad node 'm' is critical (all available paths must pass through 'm').

$TF_{\max}$ is a parameter defining the maximum number of nodes (thus, the maximum quantity of signal) for which a nomad node 'm' is critical (all available paths must pass through 'nm').

d_{ts} is a parameter representing the distance between two nodes 't' and 's'.

d_{max} is a constant representing the maximum distance separating a node i and a target j or separating two nodes i and j so that they could detect each other.

l is a parameter representing the RoI (region of interest) length.

w is a parameter representing the width of the RoI.

h is a parameter representing the height of the RoI.

Lf > 0 is the lifetime of the network (i.e. time till the required degree of coverage is maintained) and *Lfmax* is an upper bound for ***Lf***. Lf_i is the lifetime of the sensor i ∈ V.

TP_i^r representing the transmitted power of the signal (the emitted RSSI) of the sending node i∈V.

RP_i^r representing the received power of the signal (the emitted RSSI) at a distance *r* from the sending node *i*∈V.

3.2.1.4 The objective function

The many-objective fitness function is: Maximize F($\vec{x}$) where F($\vec{x}$) = (*f1*, ... , *f14*).

3.2.2 The details of the objectives

3.2.2.1 The number of the added nomad nodes

The number of nomad nodes to be added must be minimized. The following function is proposed for the number of added nomad nodes:

$$f1 = \text{ Minimize } \sum_{s \in Sb} Pnd_s^n \quad (1)$$

Subject to

$$\sum_{s \in S} Pfx_s^n \leq nbF \ \ \forall s \in S, n \in N \quad (2)$$

$$\sum_{s \in S} Pnd_s^n \leq nbN \ \ \forall s \in S, n \in N \ (3)$$

$$\sum_{s \in S} Pmb_s^n \leq nbT \ \ \forall s \in S, n \in N \quad (4)$$

3.2.2.2 Energy consumption

A deployed active sensor dissipates energy when transmitting, sensing, receiving, or being idle. Therefore, energy efficiency is considered as a fundamental key in designing a wireless sensor network. Because sensing and being idle energies are negligible compared to the receiving and transmitting energies, we proposed a model where E_i^{elec} represents the dissipated energy to activate the transmitter/receiver circuit and $\in_{amp}$ represents the transmitter amplifier to communicate. The energy consumed to transmit an m-bit packet to a distance d is E_i^{transm} and the energy consumed to receive the same packet is E_i^{recv} .

f2 = Minimize $\sum E_i^{transm} + \sum E_i^{recv}$ (5) where $E_i^{recv} = E_i^{elec} \times m$ and $E_i^{transm} =$

$E_i^{elec} \times m + \in_{amp} \times m \times d^2$ (5) Subject to the following constraints:

In order to minimize the consumption of energy, we can minimize the interferences during transmission. The interference caused by neighboring nodes can be reduced by limiting the maximum number of neighbors that a sensor/relay can have as shown in (5)

$$\sum_{s=0,n=0} Pfx_s^n + \sum_{s=0,n=0} Pnd_s^n + \sum_{s=0,n=0} Pmb_s^n \leq nbT + nbF + nbN + |S| - |S| \times (Pfx_s^n + Pnd_s^n + Pmb_s^n) \quad (6)$$

$$0 \leq Bt_i \leq E_0 \leq E_i^{transm} + E_i^{recv} \forall i \in V \quad (7)$$

Constraint (7) implies that the expenditure of each sensor in energy cannot exceed the available energy in the battery of this sensor.

$L_{ss'}^{ab} \leq Z_a^k$, ∀ a, b, s, s' ∈ V, k∈ K (8) where Z_a^k is equal to1 if the sensor ***a*** is activated during a period k ∈ K.

Constraint (8) indicates that if there is a route passing through the sensor '***a***', then '***a***' should be in a period of activity.

3.2.2.3 Hardware deployment cost

WiNo nodes **(Van den Bossche et al. 2016)** support the IEEE 802.15.4 protocol and represent a practical solution for indoor generic sensing nodes. The nomad WiNo nodes to be added can be of different heterogeneous types ($n \in N_b$). Even if they are all of homogeneous types, the cost of deploying the same node varies according to the site ($s \in S_b$). For example, deploying a node attached to a wall is considered cheaper than attaching it in the middle of room space. Thus, the deployment cost can be considered as an objective to minimize separately from the minimization of the number of added nomad nodes. Thus:

$$f3 = \text{Minimize} \sum_{s \in Sb} \sum_{n \in Nb} Pnd_s^n C_s^n \quad (9)$$

3.2.2.4 Network Utilization

In order to improve the lifetime of the network, many nodes can be placed close to the base station(s) which may result in poor utilization of the network resources and increased cost of deployment. Thus, it is important to maximize the network lifetime while deploying a reasonable number of nodes. The network utilization (NU) is defined as:

$$Maximize \;\; lf / \sum (Pfx_s^n + Pnd_s^n + Pmb_s^n), \forall\, s \in S, n \in N \quad (10)$$

To linearize our model, we suggest a new variable $\overline{lf} = 1/lf$. Thus, (10) becomes:

$$f4 = Minimize \;\; \overline{lf} \times \sum (Pfx_s^n + Pnd_s^n + Pmb_s^n), \forall\, s \in S, n \in N \quad (11)$$

Subject to $\overline{lf} \times \sum (Pfx_s^n + Pnd_s^n + Pmb_s^n) \le \overline{lf \max}$ where $\overline{lf \max} = 1/lf \max$ (12)

3.2.2.5 Localization rate

We propose a hybrid localization scheme that improves the used range-free technique (3D DV-Hop) by introducing a range based localization RSSI (Received Signal Strength Indication). To guarantee better localization, each target t ∈ T should be monitored by at least n_{min} anchor nodes. Then, $\sum_{s \in S} x_{ts} \ge n_{\min} \forall t \in T$. Thus, the following function (13) is suggested to model the localization:

$$f5 = \text{Maximize} \;\; \sum_{t \in T} \left(\sum_{s \in S} x_{ts} - n_{\min} \right)^+ \quad \text{where } (x)^+ = \max(0,x) \quad (13)$$

Subject to $\sum_{s \in S} x_{ts} \ge n_{\min} \forall t \in T$ (14)

Constraint (14) indicates that the number of nodes receiving a signal (calculated by the RSSI) from the target *i* should be greater than or equal to the minimum necessary to localize it.

3.2.2.6 Coverage rate

The coverage rate depends on the targets to cover. The FER (Frame Error Rate) is the metric used to measure the coverage degree. To guarantee a full coverage, each position in the 3D indoor space should be monitored by at least n_{min} nodes. Hence, $\sum_{s \in S} CovP_{ijk} \ge n_{\min}$. Thus, we suggest the following function (15) to model the coverage:

$$f6 = \text{Maximize} \sum_{t \in T} \left(\sum_{s \in S} CovP_{ijk} - n_{\min} \right)^{+} \quad \text{where } (x)^{+} = \max(0,x) \quad (15)$$

3.2.2.7 Lifetime

In the literature, the network lifetime can be defined as the time during which the first node totally dissipates its energy or as the time until the first loss of coverage appears. In fact, different factors, such as node density, node transmission, initial energy and routing strategies, can influence the network lifetime. To model the lifetime, we suggest the following function:

$$f7 = \text{Maximize } Lf \quad (16)$$

Subject to the following constraints:

$$Lf = \min_{i=1,2,\ldots,N_{\max}} Lf_i \quad (17)$$

where $N_{max} = Pfx_s^n + Pnd_s^n + Pmb_s^n, \forall\ s \in S, n \in N$ and $Lf_i = Bt_i / \max(E_i^{transm} + E_i^{recv}), \forall i \in V$.

The lifetime of the network is equal to the minimum lifetime Lf_i among the lifetimes of all sensors.

$$\sum_{s' \in S} L_{ss'}^{ab} \times Lf - \sum_{s' \in S} L_{s's}^{ab} \times Lf = D_s \times Lf \times Pfx_s^n \forall s \in S, n \in N \quad (18)$$

$$\sum_{s' \in S} L_{ss'}^{ab} \times Lf + \sum_{s' \in S} L_{s's}^{ab} \times Lf \le U_{\max} \times Lf \times Pfx_s^n + U_{\max} \times Lf \times Pnd_s^n \forall s \in S, n \in N \quad (19)$$

$$\sum_{s' \in S} E_i^{transm} \times L_{ss'}^{ab} \times Lf + \sum_{s' \in S} E_i^{recv} \times L_{s's}^{ab} \times Lf + D_s \times Lf \times Pfx_s^n \le Bt_i \times Pfx_s^n + Bt_i \times Pnd_s^n \forall s \in S, n \in N \quad (20)$$

$$\sum_{s \in S} Pfx_s^n + \sum_{s \in S} Pnd_s^n \le N_{\max} + |S| - |S| \times (Pfx_s^n + Pnd_s^n) \forall s \in S, n \in N \quad (21)$$

$$0 \le L_{ss'}^{ab} \times Lf \le U_{ss'} \times Lf \times Pfx_s^n + U_{ss'} \times Lf \times Pnd_s^n \forall s, s' \in S, n \in N \quad (22)$$

where:

$\boldsymbol{D_s}$ is the rate at which the information is generated at a sensor located at $i \in S$.

$\boldsymbol{U_{max}}$ is the node capacity*:* the maximum amounts of data a node can handle (receive or transmit) per unit time.

$\boldsymbol{U_{ss'}}$ is the capacity of the wireless link (*s, s'*).

3.2.2.8 Connectivity rate

If any node can communicate with any other node, the network is considered as connected. Therefore, any node must have at least one incoming and one outgoing link. In addition to the number of nodes and their density, the connectivity probability is typically related to the transmission range and the received signal strength. To model the connectivity rate, we suggest the following function:

$$f8 = Maximize\ RP_i^r \quad (23)$$

Subject to

$$RP_i^r \le T_{ss'} \times Sg_{ss'} \times \alpha \times r^{-\omega} \times TP_i^r \quad (24)$$

where ω is the path loss exponent (generally $2 \le \omega \le 5$) and $\boldsymbol{r}$ is the distance between the sending node and the receiving one.

$$r = r_c \Leftrightarrow RP_i^r = P_{\min}^n \quad (25)$$

Constraint (25) indicates that the sender can be connected to the receiver and the data can be received only when the power at the receiver is greater or equal to $P_{\min}^n$. The transmission range r_c is defined by $RP_i^r(r = r_c) = P_{\min}^n$.

As the wireless connectivity problem is generally abstracted into a graph theory problem, A WSN can be modeled as an undirected graph G(V;E). The probability of connectivity of the graph (then the network) will be: $\Pr ob^G = (1 - e^{-\lambda \pi r_c^2})^n$ where n is the number of nodes, λ is the node density and an edge exists between two nodes within a distance r_c. Hence, the transmission range of each sensor r_c must satisfy:

$$\sqrt{-\ln(1-(\Pr ob^G)^{1/n})/\lambda\pi} \le r_c \quad (26)$$

$$\sum_{s \in S} x_{ts} \le \sum_{n \in Na} Pfx_s^n + \sum_{n \in Nb} Pnd_s^n \quad (27)$$

Constraint (27) denotes the number of nodes able to detect a target. This number should not exceed the number of the installed nodes in the different sites.

3.2.2.9 Robustness and fault tolerance

Robustness and fault tolerance allow the network to continue operating correctly when a failure occurs on some of its nodes, leading to one or more faults. We assume to measure the system fault-tolerance in terms of robustness. Let $\boldsymbol{Ft_i}$ be the measure of robustness of the i^{th} communicating attempt launched by the sensor j and ***Tftj*** the total number of the failed communicating attempts launched by the sensor j. The robustness of the system can be represented as: $FT = \sum_{j \in V} \sum_{i=1}^{Tft_j} Ft_i$ (28) where a FT value approaching zero indicates a system with no robustness to faults, while a high ***FT*** value shows a highly-robust system. Thus, we suggest the following function: *f9*= Maximize ***FT*** (29).

3.2.2.10 Quality of links

The link quality indication (LQI) is a measure which allows characterizing the quality and the strength of received data packets. It is used in many protocol decisions as routing, packet retransmission and recovery as well as transmitted power strength. LQI concerns different layers. For example, the choice of routes on the network layer can be based on the reported LQI levels from the mac layer. Then, LQI is available to the application layer for analysis. Generally, paths having the highest overall LQI are better used to deliver data to their destinations. The LQI measure can be estimated using software in the transceiver as in the equation (30).

$\boldsymbol{Lqi} = (\boldsymbol{RC} - \sigma) \times \tau$ (30) where *RC* represents the correlation value which is the raw LQI value obtained from the last byte of the message. σ and τ are two empirical parameters based on the Packet Error Rate (PER) measurements as a function of RC.

f10= Maximize ***Lqi*** (31)

Subject to the constraint (32): To have an idea about the communication quality on the network, we propose to calculate, based on the routing protocol, the average communication quality (Alq) obtained by computing the mean path-loss values of direct communications between nodes: $Alq = \sum_{i=1}^{N_{\max}} \sum_{j=1}^{N_{\max}} \left| RP_i^r(i,j) \right| / \sum_{i=1}^{N_{\max}} |N_i|$ (32) where N_i is a node with an identifier i, $N_i \in$ V.

3.2.2.11 Path loss and fading

Path loss is the attenuation of an emitted signal due to the characteristics of the channel of propagation and the travelled distance. In the case of indoor wireless networks and in order to better model wireless communications, the path loss should be designed as realistic as possible by considering power constraints and signal coverage.

To model the path loss in the 3D space, we use the Volcano Indoor Multi-Wall propagation model which is based on the Cost231 Multi-Walls Multi-Floors propagation model as in **(Saunders et al. 2007)**. This model calculates the path losses by taking into account the indoor building structure. The formula for this model is as follow:

$$f11 = \text{Minimize } PL(d) \quad (33)$$

where $PL(d) = (PL_{fs}(d_0) + 10\alpha \log_{10}(d / d_0)) + PL_c + \sum_{i=1}^{I} N_w \times PL_w$ and

$PL_{fs}(d_0) + 10\alpha \log_{10}(d / d_0)$ is the path loss corresponding to the unit disk model representing the average path loss at a distance d (in dB), knowing that $PL_{fs}(d_o)$ is the free space path loss at a reference distance d_o from the transmitter and PL_c is an adjustable constant representing the loss. PL_c and $\propto$ are determined empirically from experimental measurements. ***I*** corresponds to the number of wall materials existing in the building, N_w denotes the walls number made from the ith material existing along the transmission path between the receiver and the transmitter, and PL_w corresponds to the loss associated with the signal passing through a wall made of the ith material. In order to overcome the problem of channel fading, a sufficient received power must be guaranteed to minimize the data packet loss rate.

3.2.2.12 Packet delivery cost and reliability

There is a relation between the node positions and data transmission, which implies a data delivery cost. To estimate the data delivery cost, an energy-related cost function is used. Thus, the total cost of data delivery denoted by *Dv* can be given by:

$$f12 = \text{Minimize } Dv = \sum_{i}^{N_{max}} Dt_i \times c_i \quad (34)$$

where Dt_i is the quantity of data (expressed on bits) transmitted by a sensor i and c_i designates the cost of data transmitted from any sensor *i* to another sensor.

Subject to

$RL(Dv) \geq RL_{min}$ (35) where ***RL(Dv)*** is the reliability of transmitting the packets, RL_{min} represents the minimum desired probability for a successfully packet delivery.

$AD(Dv) \leq AD_{max}$ (36) where ***AD(Dv)*** stands for the average delay for a successful received packet delivery and D_{max} corresponds to the desired maximum average delay.

3.2.2.13 Antenna orientation

The used nodes are equipped with omni-directional antennas which radiate radio wave power uniformly in all directions in the 3D space. While omni-directional antennas have no favorite position when transmitting, there will be no perfect omni-directional. Instead, there is always a direction where the transmission power is better than other directions. Indeed, to enhance transmission, the angle between the direction of antennas of the receiver and the transmitter should be as shown in Figure 3.2(a). Let:

Ag(i) ∈ [0,360[be the angle representing the orientation of the sensor i.

Angles(i,j) be equal to 1 if | Ag(i) - Ag(j)| ∈ [90,270[, which means that the orientation of the two antennas of the nodes, i and j, is favorable for emission (see Figure 3.2(c)); 0 otherwise.

Dir(i,j) is equal to 1 if i and j did not oppose directions (as in Figure 3.2(b)), means that the orientation of the two antennas of the nodes, i and j, is supposed to be favorable for emission (the phenomena of reflexion is neglected here); 0 otherwise.

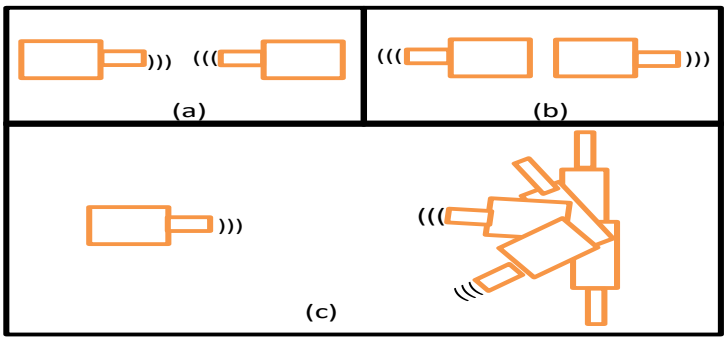


Figure 3.2 Different orientation positions of antennas

$$f13 = \text{Maximize} \sum_{i=0,j=0}^{nbF+nbN+nbT} Angles(i,j) \times Dir(i,j) \quad (37)$$

Subject to the following constraints:
The number of nodes with which a node ***i*** is properly oriented (has a favorable orientation for emission) in the 3D space can reach Pnd+Pmb+Pfx-1.

$$\{|\mathrm{Ag(i)} - \mathrm{Ag(j)}| \in [90,270[, Dir(i,j) = 1\} \Rightarrow \{\sum_{i=0,j=0}^{nbF+nbN+nbT} Angles(i,j) \times Dir(i,j) \leq nbF + nbN + nbT - 1\} \quad (38)$$

The number of nodes with which the node i is perfectly oriented for emission in the 3D space is equal to 1 (see Figure 2).

$$\{|\mathrm{Ag(i)} - \mathrm{Ag(j)}| = 180, Dir(i,j) = 1\} \Rightarrow \{\sum_{i=0,j=0}^{nbF+nbN+nbT} Angles(i,j) \times Dir(i,j) = 1\} \quad (39)$$

3.2.2.14 Distance between chromosomes

In many-objective optimization, many searches proved that to improve the crossing, two neighbors should be crossed. Thus, the distance between the chromosomes to cross should be minimized.

$$f14 = \text{Minimize } \phi(i,j) \ \forall \ i \in V, j \in V \quad (40)$$

i and *j* are two candidate sensors representing the two chromosomes to cross and ϕ is the distance (in the search space) between the two sensors.

3.2.2.15 Other general constraints

$$SP^{ab} = \sum_{s \in S} \sum_{s' \in S} L_{ss'}^{ab} \quad (41)$$

Constraint (41) imposes that the path between a source node *a* and a destination node *b* is equal to the sum of the connections between nodes *s* and *s'* to connect *a* to *b*.

$$\{s = s'\} \Rightarrow \{SP^{ss'} = 0\} \quad (42)$$

Constraint (42) imposes no hops (zero hops) to link a node to it-self.

$$M_s^m \leq hp_{\max}, m \in N_b, s \in S_a \quad (43)$$

Constraint (43) prevents deploying a new nomad node *m* in a specific location if the number of hops M_s^m between this node *m* and the nearest stationary node ***s*** exceeds a fixed threshold (hp_{max}).

$$Q_s^m \leq TF_{\max}, m \in N_b, s \in S_a \quad (44)$$

Constraint (44) prevents using a new nomad node *m* in a specific location if the traffic passing through *m*, such that *m* is critical for this path, exceeds a fixed threshold (TF_{max}).

$$d_{ss'} = \alpha \times Sg_{ss'} \times \delta_{ss'}, \alpha \in R, s \in N, s' \in N \quad (45)$$

Constraint (45) links the distance to the power transmission of the signal between two nodes *s* and *s'*. α is a real empirically-determined coefficient.

$$(Sg_{ts} = 1) \Rightarrow (d_{ts} \leq d_{\max}) \forall t \in N, s \in N \quad (46)$$

Constraint (46) implies that if there is a signal Sg_{ts} between two nodes *t* and *s*, the distance (d_{ij}) between *t* and *s* must not exceed the pre-defined maximum distance (d_{max}).

$$\delta_{ss'} \sum_{n \in N} TP_s^r \sum_{n' \in N} Sg_{ss'} \geq \sum_{n' \in N} P_{\min}^{n'} \sum_{n \in N} Sg_{ss'} \quad (47)$$

Constraint (47) defines the power of transmission when a node *s* detects another node *s'*. This power should be higher than the minimum necessary transmission power so that *s'* detects *s*.

$$\sum_{n \in N} P_{\min}^n (Pfx_s^n + Pnd_s^n) \leq \delta_{ts} P_t, \forall\ \mathrm{s} \in \mathrm{S}, \mathrm{t} \in \mathrm{T} \quad (48)$$

Constraint (48) is based on the assumption that if a node installed at a site *s* is receiving a signal from a target at the position *t*, this signal should have a power greater than or equal to the minimum power required to detect the node.

$$\left\{ \begin{array}{l} \text{if s=a: } \sum_{s' \in S} L_{ss'}^{ab} = \sum_{s' \in S} L_{s's}^{ab} + 1 \\ \text{if s=b: } \sum_{s' \in S} L_{ss'}^{ab} = \sum_{s' \in S} L_{s's}^{ab} - 1 \\ \text{otherwise: } \sum_{s' \in S} L_{ss'}^{ab} = \sum_{s' \in S} L_{s's}^{ab} \end{array} \right\} \quad (49)$$

Constraint (49) concerns the number of hops between nodes according to their positions.

$$(Sg_{ss'} = 1) \Leftrightarrow (\sum_{s \in S} L_{ss'}^{ab} > 0) \quad (50)$$

Constraint (50) indicates that the existence of a used position *s* is equivalent to the existence of a quantity of information which crossing (*s*, *s'*).

$$L_{ss'}^{ab} \leq Sg_{ss'} \quad (51)$$

Constraint (51) shows that the existence of a connection (*s*,*s'*) between two nodes (to manage the traffic between a node source *a* and a node destination *b*) requires that the node installed at the position *s* receives a signal from the node installed at the position *s'* with a sufficient power of transmission.

3.3 Hybridizations and modifications on the optimization approaches for the 3D deployment

In what follows, we will illustrate the hybridizations and modifications on the optimization approaches that we suggest in order to achieve a better deployment in a many-objective case (a number of objectives exceeding three).

3.3.1 Chromosome coding for the proposed MaOEAs

For all EMOs, the chromosome coding must be specified. Indeed, a 3D position of a node is represented by a chromosome indicating the potential locations of nomad nodes in the RoI. A point (X, Y, Z) models this position. Each gene in the chromosome represents a binary digit resembling the position's value on the X, Y and Y axes. For instance, Figure 3.3 represents the chromosome corresponding to the node which is mapped to the (46, 53, 34) location. Different factors influence choosing the chromosomes population size. The most important ones are the network configuration and the RoI. For example, considering that the node radius is equal to 9 meters and the sensing area is equal to 70×80×120 meters, the number of needed

fixed nodes to deploy can be equal to 661, (because (70×80×120)/(4×π×92)=660,19 ~=661), then the initial population should be equal to 661 chromosomes randomly disseminated in the coverage area. This value is calculated assuming that 661 sensor nodes can ensure the coverage of the entire RoI in the case of a uniform deterministic deployment in the 1-coverage case (each target must be monitored by one node at least). In the case of k-coverage, the initial population to start with should be equal to 661 × k chromosomes.

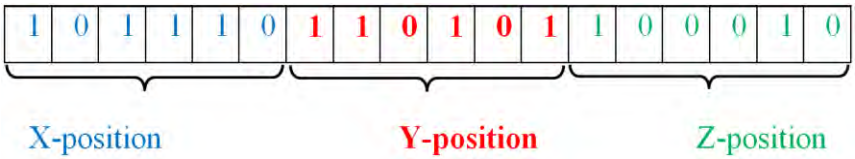


Figure 3.3 The chromosome representing the sensor in the 3D position (46, 53, 34)

The choice of the binary coding is justified by its easiness of use and its low computational cost (a low complexity) which is required when resolving MaOPs. Another reason is related to the use of the neighbourhood in recombination and mutation (as explained in the paragraph below) in our approach: Indeed, the binary coding allows better than other coding manners, to assess the differences in genes between two chromosomes. Thus, a better comparison between chromosomes according to their distances from each other will be possible. Nevertheless, the binary coding may lead to non-feasible solutions (impossible solutions). These solutions will be penalized by a weighting coefficient and will not be selected by the algorithm afterwards.

Next, the used EMOs are detailed. So are the modifications proposed to enhance these algorithms in order to allow them properly handling MaOPs.

3.3.2 Inclusion of diversity

3.3.2.1 Neighbourhood restriction and adaptive multi-operators

In MaOEAs, due to the high dimensional objective space, the population diversity increases and mutation and crossover operators becomes inefficient and may create an offspring which may be not selected as parents. To overcome this problem, we implement a mechanism based on two strategies: an adaptive multi-recombination operator's with neighbourhood restrictions (named AxN) and an adaptive multi-mutation operator's strategy with neighbourhood restrictions (named AmN).

*** Principles of the mutation and recombination with the neighbourhood (AxN and AmN)**

- The neighbourhood restriction concept

As example of the utility of using the neighbourhood in the operator's variation in MaOEAs, it is shown in (Ishibuchi et al., 2015) that MaOEAs can often apply effectively recombination to solutions having relatively similar gene structure where there is a high dependency between objectives. Thus, the neighbourhood is used aiming at improving the effectiveness of the mutation/recombination operators by increasing the number of objectives. This helps to reduce dissimilarities between new individuals since recombining (and mutating) individuals which are too distinct may be penalizing and could affects the efficiency of the operators. To achieve this, the proposed neighbourhood concept computes the distance between individuals in the objective space. Then determines the set of Nh×|P| nearest neighbours for each individual where |P| is the individual's population and Nh is the neighbourhood size (Nh = |P|×0.1 in our study). Moreover, our proposed strategy facilitates multiple convergences by permitting higher exploitation of the move-guiding areas.

- Neighbourhood Crossover (xN): In EMOs, the crossover operation allows the generation of good individuals as an offspring from parents. Ideally, this offspring must be composed of non-dominated solutions which are uniformly scattered in the population. Initially, the idea of using selection scheme mating supposes that each couple of individuals from the current population can be chosen as parents. Among the drawbacks of such mating scheme, the random choice of individuals and the large Euclidian distance between individuals in the variable space. As a consequence, the obtained solutions are more probably to be dominated. As a solution to this problem, some studies suggest a more determinist selection scheme based on the idea of considering the proximity and picking closer individuals to achieve the recombination which is very interesting for several multi-objective and many-objective problems. Thus, we propose a neighbourhood crossover that selects individuals having short Euclidian distance in the objective space so that the search ability can be reinforced by crossed individuals that are close to each other in the objective space. When crossing adjacent individuals in the variable space, the obtained offspring is generated near parent individuals in terms of their objective values and may be a non-dominated solution, which increase the probability of having population with great diversity. Although, the Euclidian distance in the variable space may not be defined in several cases like combinational functions. In the case of continuous functions, adjacent individuals in the objective space have often a high probability to be adjacent in the variable space. Thus, in our study, crossover is performed on adjacent individuals in the objective space instead of the variable space. This neighbourhood crossover algorithm is shown in Algorithm 3.1.

Input: A set of solutions (population) composing the current generation
Output: A of solutions (population) composing the next generation
01: Classify the population according to their closeness from the best individual for one of the function values in the objective space.
02: Switching the sorted individuals in a random way according to a parameter *neigh* controlling the adjusted neighbourhood with a reasonable width of the population size.
03: Choosing two adjacent individuals from the population for performing the crossover.

Algorithm 3.1 The Neighbourhood crossover algorithm

Step 2 allows escaping from local optima by the switching operation that guarantees the non-conducting with the same pair in every generation. The *neigh* parameter controlling the width of the population size is a percentage that represents the ratio of the size of the population. Thus, if the value of neigh is set to 10, the adjusted neighbourhood is conducted using a population width which is equal to 10%. As a consequence, the proximity of individuals is inversely proportional to the neigh parameter. Although, increasing too much the proximity among individuals may increase the probability of repeatedly conducting the crossover into the same pair.

- Neighbourhood mutation (mN): Same as the neighbourhood crossover, the neighbourhood mutation aim at restricting the production of solutions within the same niche (local area) as their parents which imply inducing a stable niching behaviour. In our study, we aim to minimize ϕ (i,j) $\forall$ i∈V, j∈V where i and j are two candidate sensors to cross, and ϕ is the distance (in the search space) between the two sensors. To perform the neighbourhood mutation, only one parameter is needed, which is the neighbourhood size *ns*. This parameter specifies the number of members to be considered as mutation vectors in each subpopulation. In this context, the authors in (Qu et al., 2012) investigated the effect of varying neighbourhood size on the behaviour of the algorithm. Their works prove that the preferred range of the neighbourhood size is between: 1/20 and 1/5 of the overall population. For this reason, the neighbourhood size is considered as a special niching parameter which is easy to

choose since it may be taken proportionally to the population size. Thus, as evidenced by our experimental results, the neighbourhood size does not affect the efficiency of the algorithm. Among the advantages of this strategy is that it guarantees evolving each individual toward its nearest optimal point. Another advantage is the performance of the algorithm which is not dependent on the variation of the neighbourhood size. Also, according to (Qu et al., 2012), the neighbourhood mutation improves the detection of local optima. The proposed neighbourhood mutation algorithm is illustrated in Algorithm 3.2.

Input A set of solutions (population) composing the current generation
Output A set of solutions (population) composing the next generation
01: **For** each individual *i* in the population size (*N*) **do**
02: Compute the Euclid distances between *i* and other individuals in the population.
03: Create a subpopulation *sp* from the *m* nearest individuals to *i*.
04: Create an offspring *o* using the adaptive mutation strategies applied on *sp* and readjust out-bounded solutions if exist.
05: Apply the fitness function to evaluate produced offspring *o*.
06: **Endfor**
07: Create the next generation by applying the niching strategy to choose the *N* fittest solutions

Algorithm 3.2 The Neighbourhood mutation algorithm

Indeed, starting from a population of solutions of the current generation, the proposed neighbourhood procedure calculates the Euclidean distance between i individuals on the population. Then selects the n members having the smallest Euclidean distance to the individuals *i*. Afterward, an offspring is produced and evaluated using the fitness function, as a population of solutions for the next generation.

- The adaptive multi-operators concept

There is another problem confronted when MOEAs are used to resolve many objective real-world problems. This problem is the choice of the appropriate recombination and mutation operators for each problem. In our proposed strategy, the operator variations are applied adaptively. The contribution of each operator is taken into account. Indeed, the operator which succeeded in the last iteration is used to adjust the selection probability of this operator. Hence, each operator has a selection probability in the next generation which is relative to its contribution. In the adaptive mutation, the mutation probability is modified while the algorithm is executed. In our case, we use a directed adaptive mutation (Korejo, 2011), which uses the feedback information from the previous generation without modifying the probabilistic nature of the mutation. Thus, new solutions are deterministically generated in the search space and are guided toward the optimum by earlier individuals.
The proposed AxN strategy is based on a crossover with neighbourhood operation which can be performed on a pair of parents after the selection step in the EMO algorithm. Indeed, we propose to use an adaptive multi-operator recombination operator which allows the improvement of the search and adapt it to the local characteristics of the problem. The AmN strategy is based on a mutation with neighbourhood operation which is used to avoid the local optima and to increase the diversity by changing the chromosomes values. The used mutation operators are chosen adaptively.

*** Implementation of the AxN and AmN strategies on the proposed algorithms**

- ∈-NSGA-II-AxN-AmN

Our proposed adaptive neighbour scheme of the selection operators of the ∈-NSGA-II stems from the selection process of the AMALGAM algorithm (Vrugt and Robinson, 2007), which

uses a set of MOEAs controlled by a master algorithm. The AMALGAM algorithm measures the contribution of each method in the previous iteration. Then, these methods are admitted according to their contributions rates exhibiting the most relevant reproductive success. Stemming from (Nebro et al., 2013), the pseudo code of our proposed ∈-NSGA-II-AxN-AmN is given in Algorithm 3.3.

```
Input: s // the population size
Ouput: Population
01: Population ← Random Population()
02: AuxiliaryPopulation ← ∅
03: while (termination condition is not fulfilled ) do
04: for i ← 1 to (s) do
05:     Parent←Selection_two_neighbor_parents(Population);
06:     ProbaRecombinationOp ← choosing_operator()
07:     ProbaMutationOp ← choosing_operator()
08:     offspring ← Neighborhood_Adaptive_Mutation_AmN (Parent, ProbaMutationOp);
09:     offspring ← Neighborhood_Adaptive_Recombination_AxN (Parent, ProbaRecombinationOp);
10:     Evaluate Fitness(offspring);
11:     Insert(offspring, AuxiliaryPopulation);
12: end for
13:  X ← Population ∪ AuxiliaryPopulation
14:  ∈-dominance-Ranking (X);
15:  Population ←Select Best Individuals(X);
16:  Update_archive (Population);
17:  Inject_solutions(Population);
18:  Doubling_the_population_Size(Population) ;
19: end while
```

Algorithm 3.3 The proposed ∈-NSGA-II-AxN-AmN algorithm

The procedure of calculating the probability of each operator is illustrated in Algorithm 3.4. This procedure is used for selecting both mutation and recombination operators. Indeed, considering a set (N) of different operators, the ***choosing_operator()*** procedure calculates the contribution of all those operators (lines 2-8). The procedure computes the number of solutions produced by each operator that belongs to the population P of the following generation (line 3). To avoid discarding operators generating no solutions in iteration, each operator that has a contribution which is smaller than a predefined threshold, its contribution is set to this threshold (lines 4-6). This operator can have promising contribution later in other phases of the search.

```
Input: The N operators
Output: ProbaOp
01:  TotalContrib ← 0
02:  for 1 ≤ operator ≤ N do
03:     OpContrib← solutionsInNextPopulation(operator,P);
04:     if OpContrib ≤ threshold then
05:        OpContrib ← threshold;
06:     end if
07:     TotalContrib ← TotalContrib + OpContrib;
08:  end for
09:  for 1 ≤ operator ≤ N do
10:    ProbaOp ← OpContrib / TotalContrib;
11:  end for
```

Algorithm 3.4 *Choosing_operator()* procedure

Hence, among differences between the standard ∈-NSGA-II and the proposed ∈-NSGA-II-AxN-AmN, the choice of the appropriate mutation and recombination operator.

- U-NSGA-III-AxN-AmN

The third proposed algorithm is the **U-NSGA-II-AxN-AmN** which is also based on the original U-NSGA-III algorithm (Seada et al., 2016) within a modified neighbourhood mutation and recombination phase which uses adaptively all the previously indicated mutation operators. This the first time such a modification of the U-NSGA-III is proposed. Algorithm 3.5 illustrates the generation *t* of the proposed U-NSGA-III-AxN-AmN algorithm.

Input: *H* structured reference points Z^s or supplied aspiration points Z^a, parent population P_t
Ouput: P_{t+1}
01: // *Initializations identical to the original U-NSGA-III algorithm* (Seada et al., 2016)
02: P_t' = Niching_and_Neighbor_Based_Selection(P_t)
03: ProbaRecombinationOp ← **choosing_operator()**
04: ProbaMutationOp ← **choosing_operator()**
05: Q_t = **Neighborhood_Adaptive_Mutation_AmN** (P_t', ProbaMutationOp);
06: Q_t= **Neighborhood_Adaptive_Recombination_AxN** (Q_t, ProbaRecombinationOp)
07: // *The rest is the same as the original U-NSGA-III algorithm* (Seada et al., 2016)

Algorithm 3.5 The generation *t* of the proposed U-NSGA-III-AxN-AmN algorithm

- MOEA/DD-AxN-AmN

Despite the original MOEA/DD (Li et al., 2015) rely on a neighbourhood strategy, as in the previous presented algorithms, the same variations of operators are applied to the original MOEA/DD in order to take advantage of our adaptive multi-operators concept. This the first time such a modification of the MOEA/DD is proposed.

3.3.2.2 Including single-grid and mutiple scalarizing functions in the aggregation based approach (MOEA/DD)

Among the advantages of the algorithms based on scalarizing functions compared to the algorithms based on Pareto dominance, their low-cost computation and its scalability to MaOPs. Different scalarizing functions exist. Choosing the appropriate scalarizing function is a relevant issue to be considered when designing scalarizing function-based algorithms since choosing the suitable scalarizing function is problem-dependent (Ishibuchi 2010). For example, according to Ishibuchi (2009), the weighted sum is generally used when the PF is convex, but it is not suitable for non-convex PFs. The weighted Tchebycheff is often used when the PF is non-convex, but its efficiency could be affected by the increase of the objectives number. Hence, it is interesting, to adapt the MOEA/DD, in order to have the ability to automatically choose between several scalarizing functions for each individual in each generation. Authors in (Ishibuchi 2010) used two ideas for simultaneously using multiple scalarizing functions in a single MOEA/DD algorithm. The first idea is the use of several scalarizing functions in a multi-grid scheme where each scalarizing function has its unique weight vectors complete grid. The second idea is alternately assigning a different scalarizing function to every weight vector in a single grid. Their results showed that, for 0/1 knapsack problems with six-objectives, simultaneously using the weighted Tchebycheff and the weighted sum in MOEA/D outperforms their individual use. However, the number of used scalarizing functions is limited to two. Since it is easier to generalize the single-grid implementation (the second idea), our idea is to use the second implementation based on simultaneously using several scalarizing functions, with more than two scalarizing functions. Indeed, as opposed to the original MOEA/D having a single complete grid with up to15 weight vectors, our single-grid implementation scheme based on multiple scalarizing functions suppose that each weight vector has a different scalarizing function. The following scalarizing functions are considered: The weighted sum (WS) (Ishibuchi et al., 2009), the weighted Tchebycheff distance (TCB) (Ishibuchi et al., 2009), the penality-based boundary intersection (PBI) (Sato et al., 2009). Other scalarizing functions may be considered such as

the Inverted PBI scalarizing function (iPBI) (Sato et al., 2014) and the vector angle distance (VA) (Tan et al., 2013). The proposed algorithm that uses multiple scalarizing functions is named mMOEA/DD-AxN-AmN.

3.3.3 Incorporation of Dimensionality Reduction (the NL-MVU-PCA algorithm)

3.3.3.1 Incorporating L-PCA and NL-MVU-PCA on the tested EMOs

To integrate L-PCA and NL-MVU-PCA on the proposed EMOs, we apply the reduction using the offline linear and non-linear objective reduction algorithms (named respectively L-PCA and NL-PA-MVU).

In order to minimize the effect of noise and dependencies between objectives, the PCA method projects a data D on the eigenvectors of its correlation matrix while preserving its correlation structure. Indeed, PCA method removes the higher order dependencies in the given data D. Thus, PCA may become unable to capture the data sets having structures with non-Gaussian or multi-modal Gaussian distributions (Shlens et al. 2009). In fact, different nonlinear dimensionality reduction approaches, like Graph-based ones (Saul et al. 2006) use a standard kernel function to transform data. Then, they apply PCA in the transformed kernel space. Although, its success is depending on the a priori chosen kernel. In our work, we employ the PCA method, proposed in (Saxena et al., 2013), which overcomes this limitation by deriving the "data-dependent" kernels.

The second used machine learning-based method is the MVU (Weinberger et al. 2006) relying on a graph that computes the low-dimensional representation in order to unfold the high-dimensional manifold data. To perform the unfolding process, Euclidean distances between data points are maximized, while angles and distances between nearby points are locally preserved. Theoretically, this can be modeled as a semi-definite programming problem (SDP) (Weinberger et al. 2006) where the output is the kernel matrix representing the kernel space to which the PCA method is applied.

In our work, we use the framework proposed in (Saxena et al. 2013). In fact, Given an M objective optimization problem with a set of non-dominated solutions, the proposed framework aims at specifying the smallest set of m conflicting objectives ($m \leq M$) while preserving the correlation structure among the given solution set. To perform this, the proposed framework is used to eliminate globally correlated objectives and non-conflicting ones along the eigenvectors of the correlation/kernel matrix. Thus, found solutions are estimated as good representative of the PF if there is conformity between the correlation structure of the PF and that of the given

non-dominated solutions. Therefore, essential objective set includes the smallest set of conflictual objectives determined by the framework. This framework is employed to reduce iteratively the objectives until obtaining the same objective set deduced as essential in two successive iterations. More details of this framework were discussed in (Saxena et al. 2013).

3.3.3.2 Integrating PI-EMO-PC, Knee points and NL-MVU-PCA on the proposed EMOs

In order to overcome the previously-mentioned difficulties of EMOs in resolving MaOPs, we suggest a justified hybrid scheme incorporating different approaches. This scheme is illustrated in Figure 3.4.

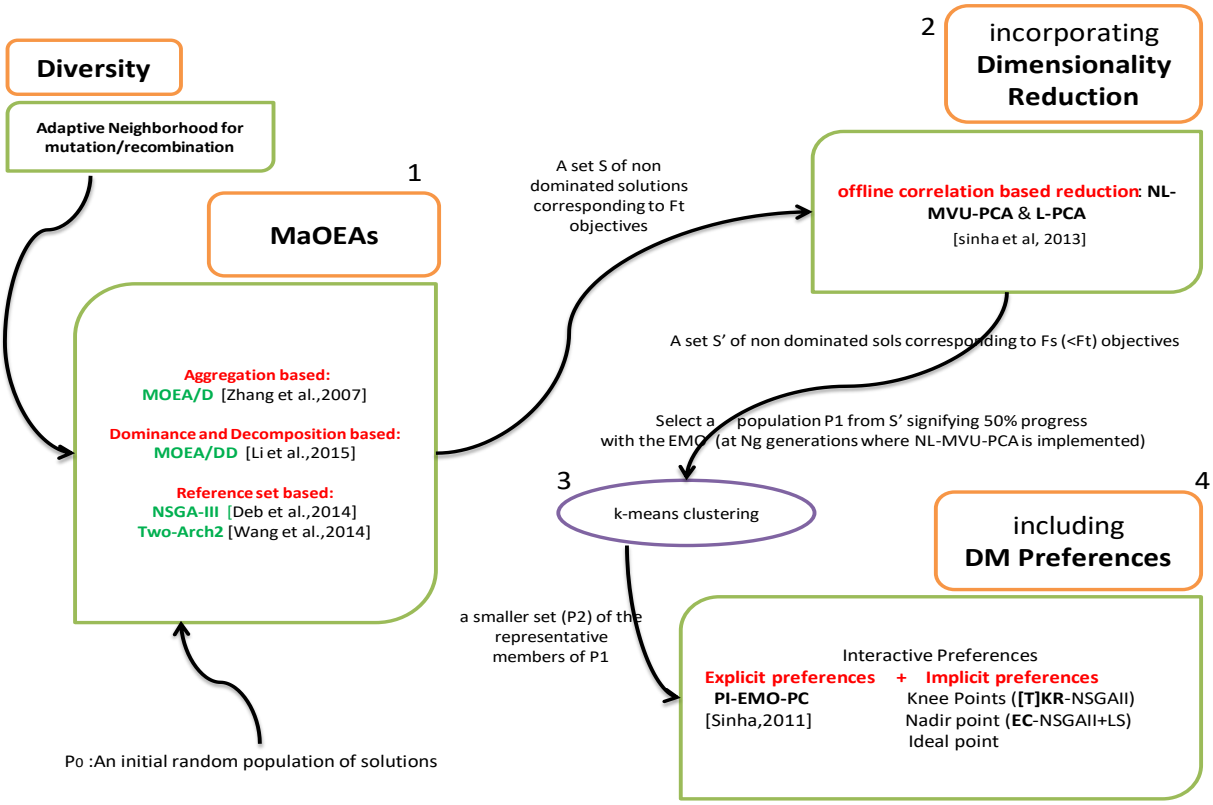


Figure 3.4 The four steps of the proposed hybridization scheme

In this scheme, four classes of MaOEA are combined: aggregation-based, reference point-based, reduction based and preference-based.
Firstly, the MaOA is executed (MOEA/DD, NSGA-III or Two-Arch2) with preserving the diversity using an adaptive neighborhood mechanism. Thus, a set of solutions corresponding to the optimization of the initial set of objectives is obtained. On this set, the dimensionality reduction is performed and a set of solutions corresponding to a smaller set of objectives is provided. Afterwards, the preference procedure is achieved.
Indeed, the preference approaches suppose that no redundant objectives exist in the given problem (Saxena et al., 2013). Thus, in our approach, dimensionality reduction procedure is always performed before applying the decision maker (DM) preferences.
Since the PI-EMO-PC procedure requires a sufficient search window, its input population should be chosen in such a way that the search converges to a solution in accordance with the interactively elicited DM's preferences. To choose the input population for our preference procedure guaranteeing a balance between the computational efficiency and the convergence to a single solution, we follow the same steps as indicated in (Sinha et al., 2013): The input population is an intermediate population taken at Ng generations (50% of progress) after applying the reduction. After identifying this input population for the preference procedure, only a smaller set representing its members is considered using the k-means clustering. These members serve as the initial population members of the PI-EMO-PC.

3.3.4 Incorporation of preferences

When resolving a problem, MaOAs gives a set of non-dominated well-sparse solutions that approximate the whole Pareto Front (PF). In the case of real world problems, the user/decision maker (DM) generally needs a small number of non-dominated solutions. Besides, the number of needed solutions and the size of the population are exponentially dependent on the number of objectives (Ishibuchi et al., 2009). Hence, it is relevant to

concentrate the search on a set of specific regions guided by the user preferences. Based on the time of incorporating the preference ordering, three main classes of preference algorithms exist (Jaimes et al., 2011): A priori, interactive and a posteriori algorithms. The second class based on interactive process is the most relevant since it permits a progressive engagement of the DM preferences and allows the readjustment of the decisions during intermediate generations of the algorithm. However, few studies focus on following the evolving of the new solutions to progressively incorporate the user preferences. In this paper, we are interested in proposing a novel hybrid scheme relying on hybridizing the NSGA-III with an interactive preference algorithm (PI-EMO-VF (Deb et al., 2010)) where the DM is asked to guide the search to a specific subset of the PF by interactively incorporating his preferences.

3.3.4.1 PI-NSGA-III-VF (NSGA-III with interactive preferences)

PI-EMO-VF (progressively interactive EMO value function) uses a generic procedure which can be incorporated to any evolutionary multi-objective optimization (EMO) algorithm. PI-EMO-VF uses an approximate value function which is progressively generated. Thus, after every few number of iterations (generations) of the used EMO algorithm, a set of fairly distributed non-dominated solutions is identified and the DM is asked to give his preference informations about the relation between solutions. In an ideal scenario, the DM provides a complete ranking of the solutions (from the best to the worst). Nevertheless, the DM can gives a partial preference-information. This information helps establishing an increasing polynomial value-function. Then, according to this value-function, the stopping condition is set up. The PI-EMO-VF algorithm is detailed in (Deb et al., 2010). Figure 3.5 illustrates the flowchart of the proposed PI-NSGA-III-VF procedure.

The solution encoding:

a) Solution representation: A NSGA-III candidate solution (chromosome/individual) is a sequence of 3D coordinates representing the nodes disseminated on the 3D indoor space. To represent this solution, a vector-based representation is used. Each dimension of the vector represents an axis of the node coordinates corresponding to its position in the indoor space. For each position, pre- and post-conditions are set to guarantee the feasibility of the solution (its applicability in the indoor space). The initial population is obtained by randomly creating a set of individuals.

b) Variation of the solution: In NSGA-III, the variation operators (recombination and mutation) help the search to move toward the optimal solutions. To perform the recombination, a SBX (simulated binary crossover) operator with large distribution index is used. To enhance the search process in the case of many-objective problems, it is recommended to create an offspring that is a neighbor of its parents (Li et al., 2015). Thus, a neighborhood strategy relying on the control of the cutting point of the recombination operator is used. This cutting point is taken either in the first tier of the vector or in its last tier. The mutation operator is based on a bit-flip operator. More details (probability, index, and operators) on the used mutation and recombination are given in section 4.7.1.1.

c) Evaluation of the solution: Once the offspring is created, the deployment solution is assessed using the metrics of the objective functions (see *f1* to *f14* in section *3.2.2*) Based on these values, a non-domination rank and a position in the objective space are attributed to the solution which allows it to be assigned to a particular reference point based on a distance computing.

d) Normalization of the population: Since the metrics of the objective functions have generally different scales and are incommensurable, a normalization procedure is used in NSGA-III to overcome this problem. Indeed, in each iteration the upper and lower bound values for each metric are retained to use them by the normalization procedure. This latter

procedure permits population members and reference points to have the same range, which is essential for preserving diversity.

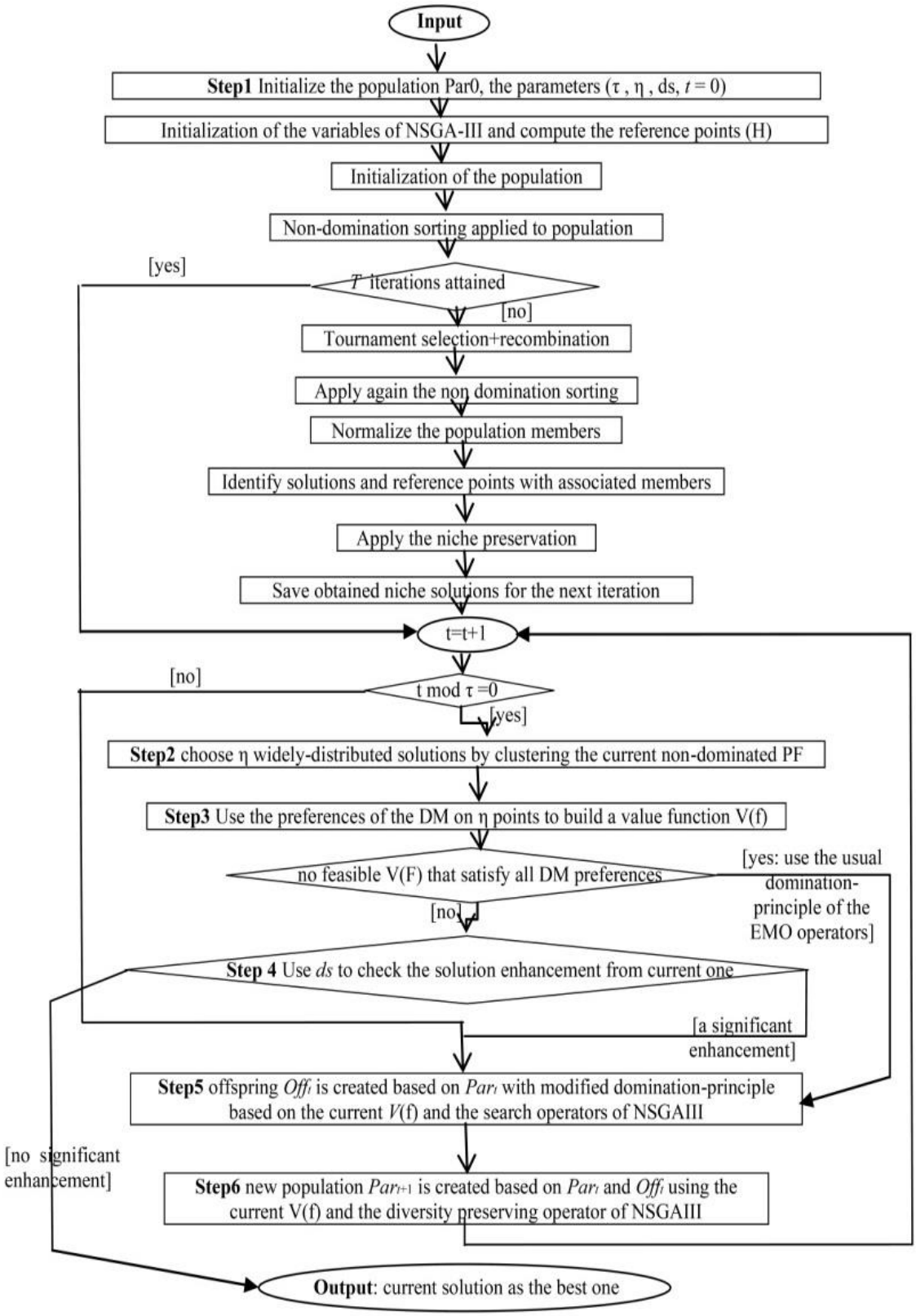


Figure 3.5 Flowchart of the PI-NSGA-III-VF (Mnasri et al., 2018a).

3.3.4.2 The proposed hybrid preference algorithm PI-EMO-PC-INK: (PI-EMO-PC-Ideal point-Nadir point-Knee points)

Interactive preferences are the most interesting since they are dynamically injected into the selection process in order to continually guide the search for appropriate actions. Figure 3.6 illustrates the proposed hybrid method based on combining implicit and explicit interactive preferences. Firstly, Ideal point and Nadir one are found respectively by using the Extremized-Crowded NSGA-II algorithm (EC-NSGA-II) (Deb et al. 2006) and by minimizing each objective individually in the search space. Then, an explicit algorithm based on PI-EMO-PC is executed if the DM has preferences. Otherwise, an implicit preference process aiming at finding knee regions based on Trade-off-based KR-NSGA-II (TKR-NSGA-II) (Bechikh et al. 2011) is carried out. When using explicit algorithm based on PI-EMO-PC, if the DM is not satisfied but he becomes tired, the procedure performs the process to find the previously-indicated knee regions. Afterward, if the DM is not tired yet, he modifies the aspiration levels of the PI-EMO-PC in order to incorporate new information about his preferences. These processes (PI-EMO-PC and TKR-NSGA-II) will be repeated if the DM is not satisfied or the maximum allowed number of permitted interventions is not reached. In both cases, when the DM becomes satisfied or the maximum allowed number of permitted interventions is reached, the global process will be stopped.

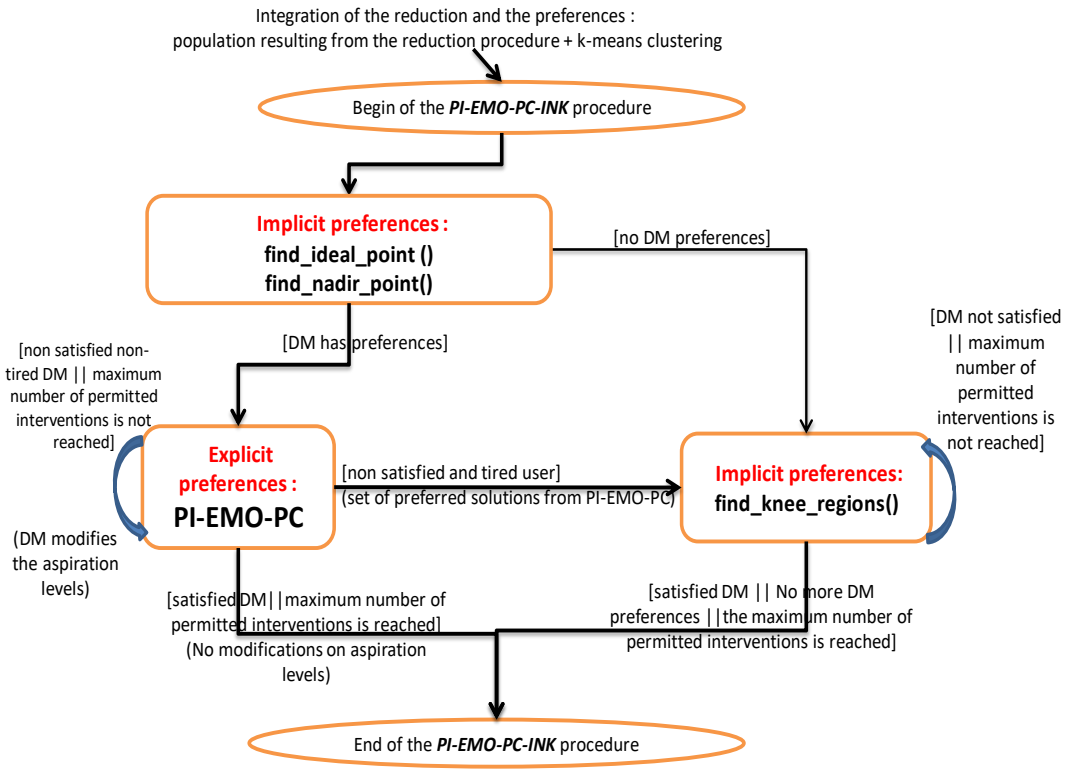


Figure 3.6 The proposed hybrid preference algorithm (PI-EMO-PC-INK)

The ideal point is a form of implicit DM preferences. It can be defined as the vector $z^I = (z_1^I, ..., z_M^I)$ composed of the best objective values over the search space Ω. The ideal point may be specified by individually minimizing each objective in the search space. Mathematically, the ideal objective vector is given by $z_m^I = Min_{x \in \Omega} f_m(x), m \in \{1, ..., M\}$ (52).

The nadir point is another form of implicit DM preferences. It can be defined as the vector $z^N = (z_1^N, ..., z_M^N)$ including the worst objective values over the Pareto optimal set. Mathematically, the nadir objective vector is given by $z_m^N = Max_{x \in P*} f_m(x), m \in \{1, ..., M\}$ (53). According to (Branke et al. 2008), several interactive algorithms used the nadir point as a pre-requisite. However, estimating accurately the nadir point for many objective problems is an open research issue. In order to help the DM in expressing his preferences, Nadir point is applied so that each aspiration level will lies between the nadir value and the ideal one.
Nadir point and ideal point assists the DM in expressing his preferences by identifying the range of the objective functions at the Pareto optimality stage. Both ideal and nadir points are employed to visualize the optimal Pareto front which facilitate comparing solutions especially for high dimension problems. Because it is used to avoid the worst instead of achieving the best, the Nadir point is considered as a more conservative point of view, compared to the use of the ideal point. Figure 3.7 illustrates the Nadir and ideal points, for a two-objective problem.

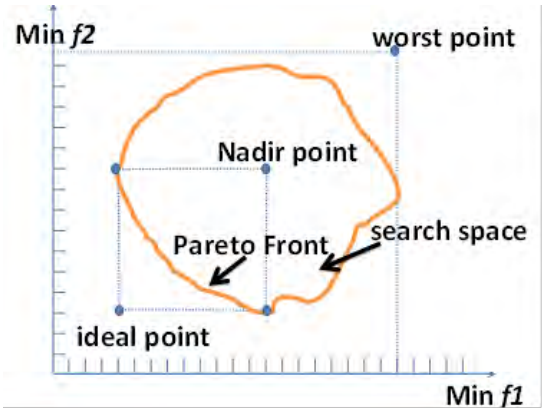


Figure 3.7 Nadir and ideal objective points

An important benefit of our approach is that it allows determining “knees” (special points in the PF where there is a maximal marginal return in the trade-off surface). In fact, knees represent points where a small improvement in the performance on one goal results in a large decrease in the performance on another conflicting goal. Due to this property, detecting such knee points is often extremely valuable and Knee points are generally interesting solutions because they allow the DM to better know and balance its conflicting internal goals.

3.3.5 AcNSGA-III: a hybrid framework for NSGA-III and Ant system

Despite its efficiency, the NSGA-III has some difficulties when solving mono-objective and two-objective optimization problems. These difficulties concern the low selection pressure that NSGA-III introduce to non-dominated solutions of a population when resolving two-objective problems. Moreover, they concern the small population size and the random selection process when resolving mono-objective problems (Ibrahim et al., 2016). Also, the ACO algorithm has a main drawback which concerns the convergence into the local optima (Sim et al., 2003).

Therefore, the idea is to propose a well-justified hybridization scheme using the two algorithms to take advantage of their strengths and to remedy their drawbacks. When hybridizing these two algorithms, most of the studies (Huang and Chne, 2013), (Shen 2016), use the standard and basic version of the genetic algorithm. Besides, most of these studies sequentially execute the two algorithms (the standard GA then the ACO or the opposite). Thus, the final solution of one of the two algorithms is the initial solution of the other. Although this basic scheme of integration enhances the ACO convergence rate, this latter remains converging excessively which makes the problem of local optimum unsolvable. In our study, we propose a

platform where the two algorithms (NSGA-III and ACO) run at the same time and interact with the same population. Thus, the ant algorithm steps are injected into the implementation of the original NSGA-III with incorporation of several modifications of the original NSGA-III. Among these modifications, the initial population of the NSGA-III which becomes the population built by the ants in the initial phase of the ACO. It should be mentioned that this is first time NSGA-III and ACO are integrated into a hybrid platform. Moreover, this is the first platform using a hybrid genetic algorithm and ACO to resolve the problem of 3D indoor deployment of nodes. The proposed algorithm, called AcNSGA-III is illustrated in Algorithm 3.6. It is a hybrid ant-Genetic algorithm which performs an ACO optimal selection of nodes, a dynamic pheromone updating and a mutation strategy. It accelerates the global convergence in order to speed up the local search which allows finding faster the suitable solutions of the 3D deployment problem. The global search ability and the randomness of the genetic operators are guaranteed which ensure conducting the operation of the genetic operator into generating routes by the ants if the ACO converges quickly. This allows to the latter finding the closing conditions and exits. Finally, since there is a low probability that ants and the NSGA-III process produce the same individual in the same iteration: the individual is added to the population unless it does not exist into it.

Input: NS (=2*N_{Pop}) size of population,
t (iteration= the number of travels per ant) = 1,
It_{max} (Maximum iteration= the maximum number of travels per ant).
Output : P_t
01: Initialize the parameters of the deployment problem (***CvD***, ***F***, ***Ctg***, ***mx***, ***Ngi0***, ***Expi***)
02: Initialize the ACO parameters (***NbA***, ***ExP***, ***ExV***, ***EvP***, ***MaxTP***, ***MinTP***)
03: P_0(Initial Population)= Each_ant_builds_a_solution ()
04: V_t =Evaluate the solutions and choose $\overline{NPop}$ feasible solutions
05: While t<=It_{max} **do**
06: Create Offspring Q_t
07: Mutation and recombination on Q_t
08: Set R_t=P_t∪Q_t
09: Apply non-dominated sorting on R_t and find $F1$, $F2$, …
10: S_t={}, i=1;
11: **While** |S_t|≤N_{Pop} **do**
12: S_t=S_t∪F_i
13: i=i+1
14: **End While**
15: **If** |S_t|< >N_{Pop} **do**
16: P_{t+1}= $\cup_{j=1}^{l-1} F_j$
17: Normalize S_t using min and intercept points of each objective
18: Associate each member of S_t to a reference point
19: Choose N_{Pop}−|P_t+1| members from F_l by niche-preserving operator
20: **Else** P_{t+1}=S_t
21: **End if**
22: Update_pheromone()
23: Each_ant_builds_a _new_solution()
24: V_t = Evaluate the solutions and choose N_{Pop} feasible solutions
25: Set P_{t+1}=P_{t+1}∪V_t
26: t= t+1;
27: End While

Algorithm 3.6 The proposed AcNSGA-III algorithm (Mnasri et al., 2017a)

The procedure ***Each_ant_builds_a_solution()*** allows to construct the candidate solutions using a model of pheromone which is a tunable probability of distribution over the space the solution. In our case a solution is a feasible repartition of nodes in the 3D space. The procedure ***Update_pheromone()*** allows the use of the candidate solutions to update the values of the pheromone in order to ensure moving towards future better solutions. As an optimization, the operation of these two procedures can be summarized and replaced by the algorithmic sequence illustrated in Algorithm 3.7.

```
01: For all the iterations i in 1:I do
02:   For all the construction steps j in 1:J do
03:     For all the ants k in 1:K do
04:       Choose and move to the next possible position of node
05:       Update local pheromone
06:     endFor
07:   endFor
08: Update global pheromone values on visited possible position of node
09: endFor
```

Algorithm 3.7 Operation of *Each_ant_builds_a_solution()* and *Update_pheromone()*

3.3.6 The proposed acMaPSO algorithm: including the concept of birds accents in MaOPSO

The proposed modifications of the standard multi-objective PSO are mainly aimed at overcoming the difficulties encountered when solving problems with several local optima. In our approach, changes are made in the swarm topology to avoid the premature character of the standard PSO algorithm: in addition to the use of the entire swarm best position (gbest), we use the best position of the local area around the particle, called the best cluster or best swarm (cbest). Moreover, we create an information link based on the concept of accent between each particle and its neighbors. These links build a graph that represents the topology of the local swarm or local community of birds. In PSO, the neighborhood of a particle represents the social structure that manages its interactions. This neighborhood may be global where each particle is in connection with all other particles or local where k neighbors are randomly defined for each particle at each iteration.

3.3.6.1 The bird accent concept

Indeed, according to recent research in biology (Oesel et al., 2017), singing birds have regional accents precisely like human beings. Indeed, the ability to sing and create a complete song that birds possess is inherited in large part from their parents. Experiments have shown that if these birds are reared in silence, they can only scream. Moreover, birds from different regions develop distinct accents. By imitating this biological concept, we propose a PSO algorithm based on a topology of accent categories of singing birds (acMaPSO). The idea is that each accent group has different parameters to accelerate convergence which improves the prevention of local optima. To evaluate the search capabilities of particles in their local areas, this algorithm rely on this new concept of accent where particles belong to different communities or groups (called clusters or swarms). To keep the diversity of the population during research, particles are separated into groups according to their accents. The particles of each accent category can select as neighbors only the least experienced particles (from their own groups or other groups).

In addition, aiming at keeping only the best particles, unlike the biological nature of birds, new bird's particles in our approach are supposed to sing better than their parents (its fitness may be better than the fitness of their parents). As shown in Figure 3.8, our approach is based on the grouping on communities of singers. Each community groups a set of geographical neighbors (parent and child birds). Bird's particles can have six categories of expertise. According to the concept of accent explained above, each bird goes through four main phases: birth, childhood (novice, crier then follower), parenthood (novice singer, singer then pro-singer) and death. Each bird's particle can choose as neighbors birds that do not have the same accent (which are not in its group) but have equivalent or smaller singing levels. Moreover, bird's particles will be penalized if their levels of expertise in singing increase, until being eliminated after they reach the maximum level. Thus, to avoid oscillations around

a local optimum, if the ***pbest*** value fails to be updated after μ iterations, the corresponding particle is penalized and its category is increased: therefore unless its current position is better than the ***pbest*** one, the bird goes to the next level (category) of singing. After φ iterations ($\mu << \varphi$; $\varphi = a * \mu$, $a > 5$), the clusters (swarms) are updated to allow the search to escape from local optima.

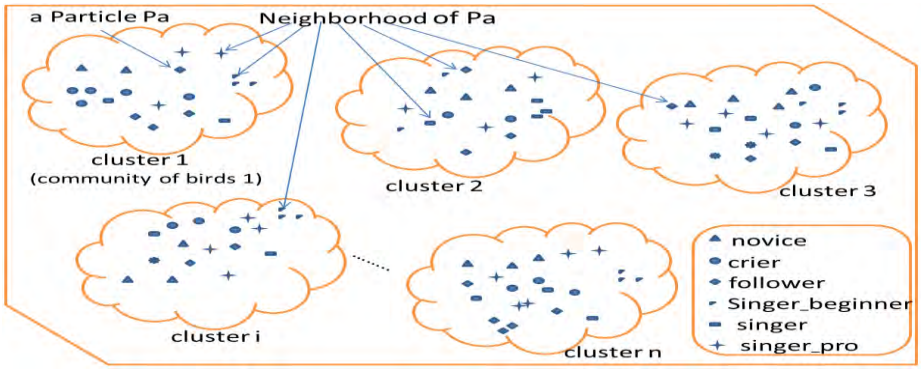


Figure 3.8 The choice of the neighborhood of a particle Pa

3.3.6.2 The accent measure

Learning accents in particles reflects the number of iterations after which the ***pbest*** position of a particle has not been updated. This latter concept can be defined as follows: $if\ f(x_i) < f(p_i)\ then\ x_i.accent = 0\ else\ x_i.accent = x_i.accent + 1$ (4.1). In fact, initially a particle is randomly generated with no singing experience and its experience improves (goes to the next level) if it cannot find a better position than its current ***pbest***. In the standard PSO, if a particle converges to a local optimum, its ***pbest*** and current position are not changed. According to the new measure of the experience of particles, if the position of a particle is not updated, its experience will be increased each μ iterations to reach the next level. This rise in the experience of the particle will continue until the particle becomes able to find a better ***pbest*** position or until reaching the final experience level and being replaced by new one.

3.3.6.3 The clustering in multiple Swarms

One of the means of enhancing the diversity in evolutionary algorithms (EAs) when resolving dynamic real-world optimization problems is considering multi-populations. In this context, we adopt a multi-swarm strategy where the population is composed of a set of subpopulations (clusters) called swarms. In multi-modal functions, one of the methods of tracking local optima is to divide the whole search space into local multi-population, which are generally covered by a small number of local optima, then separately search on these subspaces. Nevertheless, the difficulty lies in the manner of defining the area of each sub-region and the manner of guiding the particles to promising subregions. Thus, if the area of subregion is too small, the small isolated subpopulations can converge to local optima and the algorithm may do not make any progress because of the diversity lost. On the other hand, if a subregion is very large, this can cause more than one peak within the subregion of a sub-swarm.
Moreover, the optimization algorithm should have a good global search capability to explore promising subregions. To resolve theses issues, the proposed acMaPSO employs a local search method (a clustering method) to generate a proper number of sub-swarms and a global search method to detect promising subregions.
- The acMaPSO global search method: Since particles learn information from non-permanent neighbors, the population topology is dynamic. To identify (then avoid) non-promising local regions as possible, geographical neighbors are avoided if it is very close to a local optimum.
- The acMaPSO local search method (clustering method): to accelerate the convergence, local search algorithms (such as K-means, single linkage clustering or average linkage clustering)

were often combined with EAs to fine-tune the obtained solutions. Thus, clustering is used to introduce diversity and to help in avoiding local optima and non-promising local regions where no optimal solutions can be found. In our case, a hierarchical clustering algorithm (the mean linkage clustering UPGMA) is used to identify the centers of the clusters of particles in the population. The used clustering process should be efficient since it affects the overall swarm performance. In this context, in addition to the ***pbest*** and ***gbest*** parameters of the standard PSO, we add another parameter named ***cbest*** (clusterbest or cummunitybest) to keep the best position of the cluster.

Algorithm 3.8 illustrates the acMaPSO algorithm.

01: *Initialize the parameters* ***n*** *(number of swarms),* ***k*** *(population size of each swarm),*
*****w*** *(inertia weight),* ***c1****and* ***c2*** *(acceleration coefficients)*
02: *Set the fitness evaluations counter* $C_f = 0$
03: *Set the initial accent* a_i *of initial swarms to* ***novice***
04: *Random generation of* ***n*** *initial swarms* $s_i \in S$ *with size* ***P****,* ***(i*** *= 1, 2, . . . , P)*
05: for i=1 to n **do**
06: Initialize parameters of the swarm i
07: **for** j=1 to k **do**
08: Initialize position and velocity
09: Calculate fitness
10: **end for**
11: end for
12: while (stop condition is not fulfilled [sufficient good fitness | maximum number of iterations]) **do**
13: **if** highest_experience_category is reached **then**
14: Delete particles having reached the highest experience category and insert new random particles (accent operator)
15: Construct accent-group topology structure (S,a)
16: **end if**
17: **if** (fitness if not evolved in recent μ iterations) **then** (move on to the next category)
18: **end if**
19: **for** i=1 to n **do**
20: **for** i = 1 : P **do**
21: Adjust the inertial weight (S; w; a)
22: Adjust the acceleration constant (S; c1; c2; a)
23: Update position and velocity using equations **(14)** and **(15)**
24: Evaluate the fitness (S, C_f)
25: **if** (*fp* is better than *f*(*pBest*)) **then** (*pBest = p)*
26: **end if**
27: **end for**
28: $cbest_i$ = best $pbest_i$ in *P* //updating $cbest_i$ (the cbest of the swarm i)
29: **end for**
30: Update accent of particles using the equation **(16)**
31: *gBest* = best $cbest_i$ in *P* //updating $gbest_i$ (the gbest of all the swarms)
32: end while

Algorithm 3.8 The acMaPSO algorithm (Mnasri et al., 2018b)

3.3.7 Hybridizing Particle Swarm Optimization and Multi-Agent Systems

3.3.7.1 Multi-agent Systems

MAS are a decentralized system which is based on a set of agents. An agent is considered as an intelligent, autonomous and reactive entity which is able to learn and cooperate with other agents or components of the environment to achieve common goals. Each agent must be put in an environment where it can react. In general, an agent is characterized by its ability to detect its local environment in which it operates, its dependence to this environment, its ability to respond to the changes of the environment and its autonomy (ability to achieve specific tasks without external intervention).

3.3.7.2 The advantage of using the multi-agent approach in our context

The modeling of agent-based systems consists of establishing a multi-agent organization to satisfy a set of objectives. Indeed, it is a question of classifying and combining various tasks and endow the agents with a set of skills (roles and knowledge) enabling them to interact

according to a communication protocol (the contact-net, for example) to achieve their tasks. The modeling of agents concerns also the description of the complex behaviors of the agent and their relationships with other agents and with the environment. Recently, with the development of the distributed control theory, several studies are applying the theory of multi-agent systems to solve the problems of container terminals. Indeed, these agents have been used to detect disturbances on daily process plans. Several studies have highlighted the interest of MAS in solving and optimizing problems.

We propose a multi-agent approach to our problem. Indeed, MAS can reflect in a realistic (and generally efficient) way, the complex relation between the actors of a system and the way in which a solution is organized. In addition, MAS can provide a solution which is acceptable by all the actors involved in the system. The problem of scalability is also less restrictive when using the agent approach which is more appropriate for complex and dynamic systems and environments.

Indeed, the main benefit of using the multi-agent approach is that agents can learn and acquire behaviors leading to a scheduling plan that verifies and respects conflictual constraints in the case of complex and dynamic system. MAS accept and assume different players with different interests; it is a dynamic environment with disruptions.

Among the other advantages of using the multi-agent approach:

- Agents dynamically interact with the changing of the environment which makes the system robust and reliable.
- Agents, as distributed entities, are adapting to the different sizes of the problem, which allow the scalability of the system.
- Agents can access remote and geographically distributed informations.
- The multi-agent approach allows approaching a large search space which is, in most cases, not explorable via conventional resolution methods.
- We can obtain a global solution incrementally built from the partial solutions provided by the agents. Indeed, each agent seeks a locally feasible solution and negotiates its neighbors to make it globally coherent. In some complex systems that can not be modeled explicitly, each agent models a partial aspect and the overall behavior of the system is deduced from the interaction of the agent entities. Yet we often encounter the difficulty of proving the feasibility and the validity of the found solution.

3.3.7.3 The acMaMaPSO: A hybrid algorithm based-on accent multi-agent many-objective PSO

The acMaMaPSO is a new algorithm that combines the main features of acMaPSO and MAS. At first glance, PSO and MAS seem to be similar since both perform cooperative tasks and are population-based. However, particles are very distinct from agents. The first difference is that the agent can quickly and autonomously explore its environment while the particle can only move in the space of the problem following the main algorithm. On the other hand, to improve its computing performance, a particle is designed with smaller capabilities making it less intelligent than an agent relying on learning as a main component. Moreover, to simplify the design, the execution of the particles is synchronous while that of the agents is asynchronous because of the autonomy of agents.

Indeed, the hybridization of PSO with MAS combines the autonomy and learning abilities of MAS with the simplicity of PSO. As a result, particles become more autonomous, smarter and more able to take advantage of their environment. For example, if the evaluation of the fitness function is expensive, it is not necessary to perform it for each visited point and the location of this latter can be updated by communicating with its neighbors. Indeed, each agent represents a particle that reflects the behavior of a bird in its community. Each particle represents a solution in the search space and is represented by a point to which is attached a Boolean value that indicates whether this point has already been visited by an agent. If so, this

point should not be a solution. Unlike the standard PSO, the evaluation of the fitness function of the particles is not always necessary: foe example, if an agent particle arrives at a non optimal already evaluated point (marked as visited); the particle can ignore the evaluation of fitness and updates its position through the information provided by its neighbors. This contributes to the acceleration of the computation and the optimization of the performances in spite of the complexity introduced by the learning and the autonomy of the particle. In the rest of the paper, the terms particle and agent are used interchangeably.
When collaborating together, agents develop a society to achieve a common goal as well as their own individual goals. The decision-making process of the group in MAS corresponds to the fundamental nature of a particle in PSO. Thus, the proposed hybridization provides an opportunity to optimize complex problems. Nevertheless, to resolve complex optimization problems, several specificities must be defined such as the interaction method between the agents and their rules of behavior, the environment and the starting point of the research.

*** Self-Learning of agents**

In addition to their conflicting or cooperative interactions in their local environment, agents can also self-develop knowledge using their own observations and studies of the environment. According to (Wu el al., 2015), having a small range of population in an optimization algorithm positively influences its local search capability. As a result, a small range of search techniques is applied to achieve the self-learning function of agents.

*** Proposed MAS architecture and agents intentions**

The architecture of the proposed MAS is based on three types of agents: environment agent (*agEnv*), bird agent (*agBird* or *agParticle*) and swarm agent (*agSwarm* or *agCommunity*). Indeed, after initializing the parameters, *agEnv* assigns for each *agBird* a starting position in the problem space and a set of neighbors. Then, bird agents start searching for an optimal solution until a maximum number of iterations or a sufficient fitness is attained.
Firstly, each *agBird* checks if its current position is already visited. If yes, it does not evaluate its fitness function and ask its neighbors about their *pBest* and locations, then ask *agEnv* about the locations of neighboring clusters to calculate its current *pBest* and update its position and velocity. If the current position is not visited, *agBird* evaluates the fitness function, tags the current point as visited and updatse its position and velocity. The overall procedure is illustrated in Figure 3.9. For more details, see (Mnasri et al., 2018b).

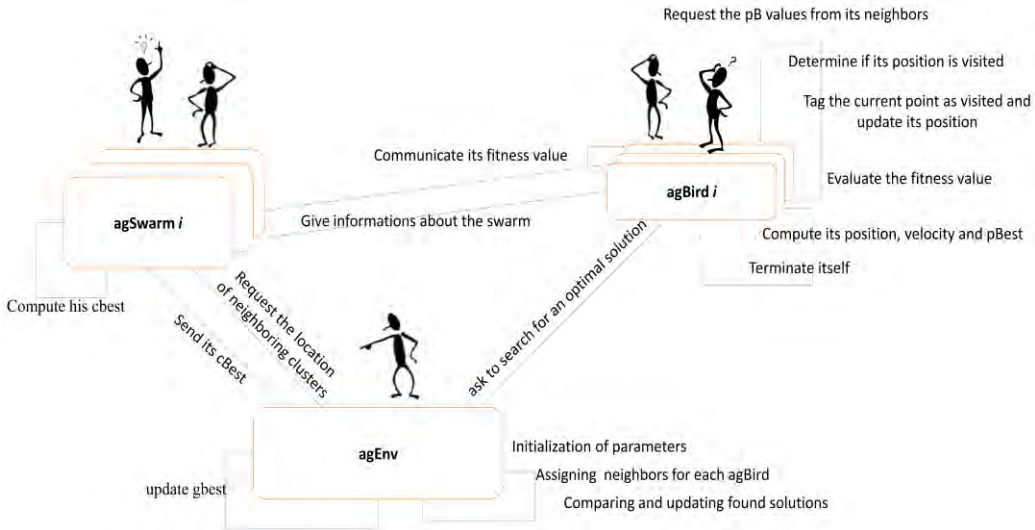


Figure 3.9 The proposed MAS architecture

The intentions are used to help the agent optimizing the value of its fitness so that he makes the appropriate response based on its environment. The fitness of the agent is determined by the equation (4.1). The details and intentions of each type of agent are as follows:

* ***agEnv***: In MAS, the design of the local environment is very relevant since agents sense the information in the local environment and act according to their strategies to achieve their intentions. The environment where the agent lives must be built so that each agent, in addition to the cooperation with its neighbors, can make self-learning and adapt its strategies of action according to its own experiences. To simplify the representation of the environment, we use a 3D cubic structure environment. Each agent occupies one point in this space and has two values that represent its speed and position. The size of the population is denoted ***L*** (a positive integer). In acMaMaPSO, the environment itself is modeled as an agent and provides other agents with additional informations (about the problem space for example). This environment agent (*agEnv*) applies a clustering algorithm (the UPGMA mean linkage clustering in our case) to discover clusters of labeled points and to provide their characteristics to particle agents (center point, density, etc). The agent *agEnv* is characterized by the following parameters:

- n which is the dimension of the problem space.
- P which is a set of points in the problem space knowing that $p \in P$,p=$\{d_1 \text{ x } d_2 \text{ x } \dots \text{ x } d_n$, *visited*$\}$ is a point in the n dimensional space and *visited* is a Boolean value indicating if this actual position is visited or not. This information helps the *agEnv* to discover clusters when applying the clustering algorithm.
- $C = \{(p_1, p_2, .., p_{center}, p_m), d \mid d \geq \in_d\}$which represents the set of clusters. p_{center} is the center point of the cluster. d is the current density of the cluster. $\in_d$ is the minimum density requirement for *agEnv* to consider a group of visited points as a cluster in the problem space. p_{center} and d can be requested by an *agBird* in its updating function to calculate its next position.

* ***agBird*** (***agParticle***): Let $Ab_{i,j}$ be the agent bird (or the agent particle) located in the position (i,j), let $N_{i,j}$ be its neighborhood, *pBest* its best position which represents, among its neighbors, the point in the problem space that is the closest to an optimal solution *gBest*. Indeed, the position of each bird agent represents a point in the search space and a solution to our deployment problem (a set of possible locations of the sensor nodes in the 3D indoor space). Instead of collecting itself informations about its cluster, an agent can simply request them from the environment agent or the corresponding cluster agent. This allows at the same time preserving the simplicity of the original PSO and significantly increasing the performance of the MAS. After the perception of its environment, each agent $Ab_{i,j}$ can perform, among others, the following tasks:

- Obtain the center location and other informations about its swarm (cluster).
- Know the status of the current position (visited or not).
- Mark a point (the current position) in the search space of the problem as visited.
- Communicate its current personal best to its neighbors.
- Request the current personal best locations of its neighbors to calculate its location.

* ***agSwarm*** (***agCluster***): This agent is responsible for synchronizing the communications between *agEnv* and *agBird* agents. Among the tasks of this agent:

- Computing the *cbest* and communicate it to *agBird* agents requesting it.
- Communicating informations about the swarm to *agBird* agents.
- It can also ask *agEnv* to provide informations about other swarms.

*** Strategies for moving agents according to the concept of accent**

Each agent is in conflict or in cooperation with its neighbors depending on its local environment and intentions. Let $L_{i,j} = (l_1, l_2, ..., l_n)$ be the set of possible locations in the range

of the optimization solution and $M_{i,j}$ = (M_1, M_2, ..., Mn) the agent having the smallest fitness between four neighboring agents. $L_{i,j}$ is considered a winner only if it satisfies the equation: F ($L_{i,j}$) ≤ F ($M_{i,j}$) (17). In this case, its position in the solution space remains the same. Otherwise, its position will change according to the equation: $l'_k = m_k + \delta.(m_k - l_k),\ k = 1,2,...,n$ (18) knowing that δ is a random number in [-1, 1]. If $l'_k < x_{k\min}$ then $l'_k = x_{k\min}$, and if $l'_k > x_{k\max}$, then $l'_k = x_{k\max}$, knowing that X_{min} is the lower limit and X_{max} is the upper limit of the possible solution space in the optimization problem. X_{min} = (X_{1min}, X_{2min},...., X_{nmin}) and X_{max} = (X_{1max}, X_{2max},....,X_{nmax}).

3.4 Conclusion

In this chapter, we gave an integer linear programming formulation as a modeling of the 3D indoor deployment problem. This model is realistic and based on a set of constraints and objectives taken from a real context of use in a deployed DL-IoT collection network. Moreover, different modifications/hybridization schemes were explained. These analysis contributions relies essentially on the use of the neighbourhood and an adaptive guided concept for the mutation and recombination operators; the use of multiple scalarizing functions in the aggregation based approaches; and the incorporation of the reduction of dimentionality, the users preferences and multi-agent systems. Aiming at improving the behavior of the tested algorithms and enhancing theirs performances in solving complex real world problems, these modifications/hybridization are assessed; in the next chapter; using evaluation metrics, simulations and real prototyping.

Chapter 4
Numerical results, simulations and experimentations on testbeds

4.1 Introduction

In the previous chapter, we presented the proposed mathematical formulation of the 3D indoor deployment problem and the different hybridization schemes aimed at improving the behavior of the tested optimized optimization algorithms and enhancing theirs performances in solving complex real world problems. In this chapter, we will first illustrate the numerical results of the 3D deployment problem after incorporating our proposed scheme of dimensionality reduction and preferences. To prove the applicability of the proposed hybridization on different problems, we will detail, in the next section, the numerical results on the DTLZ unconstrained test functions when using the same scheme of hybridization. In the fourth section, we will propose a study that presents the modeling of the used protocols then simulations with small and large instances. Afterwards, we propose experiments on real testbeds with two scenarios (with small number of nodes then with a larger number). The last two sections of this chapter will describe the obtained results when hybridizing ACO and NSGA-III (AcNSGA-III), then the results of the accent Birds PSO and the PSO-MAS algorithms.

4.2 Numerical results of incorporation of Dimensionality Reduction and of preferences on the real world deployment problem

In this section, we present the performance indicators and the parameters setting. Afterwards, we demonstrate the performance of the suggested hybrid approach on real testbeds. In our works, the implementation of the MOEA/DD, NSGA-III and Two_Arch2 in the PlatEmo framework (Tian et al. 2017) is used.

*** Performance metrics:** To evaluate the approximation sets produced by running a MaOA, several performance metrics can be employed to compare these approximation sets using numeric values. Table 4.1 illustrates the most important performance metrics used to compare many-objective algorithms.

Table 4.1 Comparative table of the most appropriate/recommended metrics for MaOAs

Performance Criteria	Metric	Expression (according to (Jiang et al. 2014))	Computational complexity	Computational comparison set	Prameter Requirements	Sets	Number of citations (according to (Riquelme at al. 2015))
Capacity (cardinality) | x | defines the cardinality or the number of elements in the set x ; t=generation index	**ONVG(S)** **GNVG** **NVA**	**ONVG(S) = |S|** **GNVG(S) =|S(t)|** **NVA(S, t) = GNVG(S,t)−GNVG(S, t−1)**	**Low**	**None**	**None**	**Unary**	**level maintained**
	ONVGR **GNVGR** **ER (Error Ratio)**	**ONVGR(S, P) =|S|/|P|** **GNVGR(S, P) =|S(t)|/|P|** **ER(S, P) = 1 −|S∩P|/|P|**	**Low**	**Pareto front (P)**		**Unary**	
	$C1_R$ **$C2_R$**	**$C1_R$ =|S∩R|/|R|** **$C2_R$(S,R) =** $\left\|\left\{\vec{s} \in S / \exists! \vec{r} \in R : \vec{r} < \vec{s}\right\}\right\| / \|S\|$	**O(m|S|.|R|)**	**Reference set (R)**	**None**	**Unary**	**level maintained**
	Metric C (Coverage of Two Sets)	**C(S1, S2) =** $\left\|\left\{\vec{s2} \in S2 / \exists\ \vec{s1} \in S1 : \vec{s1} \le \vec{s2}\right\}\right\| / \|S2\|$	**O(m|S_1|.|S_2|)**	**Optimal solution set (S)**	**None**	**Unary**	**level maintained**
Diversity (distribution and spread)	Δ^* **(S,P)**	$\Delta^*(S,P) = \frac{\sum_{k=1}^{m} d(\vec{e_K}, S) + \sum_{i=1}^{\|S\|} \| d_i - \bar{d} \|}{\sum_{k=1}^{m} d(\vec{e_K}, S) + (\| S \|)\bar{d}}$ **Where** $d(\vec{e_k}, S) = \min_{\vec{s} \in S} \| F(\vec{e_k}) - F(\vec{s}) \|$ **and** $\vec{e_k} \in P$ **is the extreme solutions on the kth objective.** $d_i = \min_{\vec{s}_j \in S, \vec{s}_j \neq \vec{s}_i} \| F(\vec{s_i}) - F(\vec{s_j}) \|$ **to identify the closest pairwise solutions in S** $\bar{d}$ **is the average of *di***	**O(m|S|^2.m|S|.|P|)**	**Pareto front (P)**	**None**	**Unary**	**Decreasing**
Convergence	$I^l_{\epsilon+}$ **(∈-indicator)**	$\inf_{\varepsilon \in R} \{\forall \vec{p} \in P \mid \exists\ \vec{s} \in S : \vec{s} \le \vec{p} + \varepsilon\}$ **where ε defines the value required to translate or scale the optimal solution set S such that S**	**O(m|S| . |P|)**	**Pareto front (P)**	**None**	**Binary**	**Decreasing**

		dominates P					
Convergence -Diversity	IGD(P,S)	$\frac{(\sum_{i=1}^{\|P\|} d_i^q)^{1/q}}{\|P\|}$ Where $d_i = \min_{\vec{s} \in S} \|\| F(\vec{p}_i) - F(\vec{s}) \|\|$ $\vec{p}_i \in$P and q=2 di is the smallest distance of $\vec{p} \in$Pto the closest solutions in S.	O(m|S| . |P|)	Pareto front (P)	None	Unary	Increasing
	Hypervolume (HV)	HV(S,R) = volume($\bigcup_{i=1}^{\|S\|} v_i$) where vi is a hypercube constructed with the reference set for each solution $\vec{s}_i \in S$	O($\|S\|^{m-1}$)	Reference set (R)	None	Unary	Increasing

where:

ONVG : Overall Non-dominated Vector Generation
GNVG : Generational Non-dominated Vector Generation
NVA : Non-dominated Vector Additional
ONVGR : Overall Non-dominated Vector Generation Ratio
GNVGR : Generational Non-dominated Vector Generation Ratio
$C1_R$: Found Ratio of the Reference Points
$C2_R$: Non-dominated Points by Reference Set

Given this variety of metrics, a research issues is questionable: What metrics should we choose when evaluating the performance of MaOPs? : Inverted Generational Distance (IGD) is less costly in calculating then the Hypervolume (HV) especially for high dimensional PFs. Yet, the comparing set 'P' in IGD may be more difficult to build than the reference set 'R' in HV. If the true Pareto front of the problem is known, the closeness to the true Pareto front can be measured. Otherwise, if no information is available about the true Pareto front, the hyper-volume and space metric can be used. Indeed, there are many metrics that are not suitable to evaluate MaOPs. For instance, we can cite the Δ metric which can be applied only for two-objective problems. On the other hand, although the HV is an expensive computationally metric, it is useful in the case of real world problems that require a prior knowledge of the real Pareto front. As a consequence, various studies (Fonseca et al. 2006), (Beume et al. 2009) were carried out to reduce the computational complexity of the HV. To sum up, the HV is necessary applied to evaluate the MaOPs because it does not require a prior-knowledge of the true Pareto-front which is important when benchmarking on real-world problems. Besides, this metric gives a single measurement to assess both the spread and convergence of solutions. However, the IGD can be used with synthetic problems due to its low computational cost.

The HV introduced in (Zitzler et al. 1999), also known as hyper-area S metric and Lebesgue measure, is a metric representing the volume of the objective space dominated by a solution set. It is a unary metric which is used to measure the size of the objective space covered by an approximation set. The HV is also the unique unary metric having the capability of measuring different aspects such as cardinality, diversity and accuracy. In the case of minimizing the objectives, smaller HV values are better used since they denote smaller difference between the approximation set and the reference set. Since it allows having an idea on both convergence and diversity, HV is a preferred performance metric but its computational cost makes it infeasible in several complex many-objective problems. In fact, it can be adapted to handle the user preferences in MaOAs as in (Wickramasinghe et al. 2010).

The IGD metric (Coello et al. 2002) is the inverted variation of the GD metric. The principal difference is that GD calculates the average distance, while IGD computes the minimum Euclidean distance between the PF and an approximation set ***A***. Moreover, the solutions in the PF are used as reference in IGD instead of the solutions in the set ***A*** to calculate the distance between the two sets. Besides, if sufficient members of the PF are known, the IGD metric will be able to measure both convergence and diversity of the set ***A*** (Riquelme et al. 2015).

According to (Jiang et al. 2014), the HV metric shows conflicting (similar, respectively) trends to IGD on the 4-dimensional concave (convex, respectively) PFs.

*** Parameters setting**: The setting of the parameters affects considerably the performance of the tested algorithm when resolving a particular problem. Yet, performing a set of

experiments using several population sizes, number of objectives, number of generations and operators is necessary when testing each many-objective algorithm separately. The objective number varies between 2 and 8, for the real world problem, and between 3 and 15 for the test functions. In all tested problems (the real world problem and the test functions), the best performance for each instance is shown with a gray background. The considered EMO algorithms have different parameters. Unless a modification in the value of a parameter to test the impact of varying this parameter, the common used values of these parameters can be summarized as follows:

- The reproduction operators: The probability of crossover is $\boldsymbol{pc}$= 0.9 with a distribution index $\boldsymbol{\eta c}$= 50. The probability of mutation is $\boldsymbol{pm}$= 1/$\boldsymbol{n}$ with a distribution index $\boldsymbol{\eta m}$= 30. $\boldsymbol{n}$ is the problem dimensionality.
- Population Size: Several specifications of the population size and the number of weight vectors for different numbers of objectives are used for each test problem.
- The number of runs: Each algorithm is performed 25 times (25 independent runs) with each configuration. Then, the various configurations are compared based on HV, in the case of the real-world problem, and on IGD in the case of the test functions.
- The termination condition is the maximum number of generations (solution evaluations) varying between 400 and 1500.
- The used scalarizing functions are: Weighted sum, Weighted Tchebycheff and PBI with a penalty parameter (θ = 0.01, 0.5, 1.0 or 5.0).
- The neighborhood size is set to 20 and the probability to select a parent from the neighborhood is 0.9.

4.2.1 Results on a constrained real world problem: the 3D Deployment in indoor WSNs

As application, we use a 3D indoor deployment WSN optimization problem with eight objectives. As described in the modeling (section 3.2), this problem has 10 decision variables as inputs and fourteen objective values as outputs. In this section, due to the complexity of computing the HV, only eight objective values to be optimized are considered. The difficulty of the tested real-world problem (3D Deployment Problem) increases with the number of objectives. Since we apply a high complexity metric (HV) to evaluate our problem, (because of the unknown Pareto front of our real world problem), the number of the used objectives ranges from 3 to 8 objectives among the objectives cited in Table 4.2. Concerning the parameters of the problem, unless indicated, the same parameters detailed in the experimentation (see section 4.5) are used: number of nodes, average number of runs and nodes repartition.

Table 4.2 Used objectives and their significance

Obj no	Signification
f1	Number of the added nomad nodes
f2	Energy consumption
f3	Hardware deployment cost
f4	Network Utilization
f5	Localization rate
f6	Coverage rate
f7	Lifetime
f8	Connectivity rate
f9	Robustness and fault tolerance
f10	Quality of links
f11	Path loss and fading
f12	Packet delivery cost
f13	Antenna orientation
f14	Neighborhood distance between parents to be crossed

4.2.1.1 Testing the effect of interdependence between objectives

The size of the population is set to 1000 (a large population) which is run for different number of generations. The used scalarizing function (for MOEA/D and MOEA/DD) is PBI (0.5). The mutation probability is 1/400 with an index of 50 (a Bit-flip mutation), and the

recombination probability is 0.9 with an index of 5 (a simulated binary crossover (SBX)). No neighbor mating in recombination and the objectives are correlated (For each experiment with N objectives, at least N/2 objectives are correlated). We employ 250 reference points for NSGA-III and MOEA/DD. In order to reduce the computational cost when calculating the HV, the improved dimension-sweep algorithm proposed in (Fonseca et al. 2006) is used to compute the HV. Tables 4.3 and 4.4 illustrate the obtained results.

Table 4.3 Best, average and worst values of HV with non-correlated objectives obtained using 15 independent runs

Obj Nbr	Max Gen	MOEA/D(PBI)	MOEA/DD	NSGA-III	Two_Arch2
3	**400**	0.989374 0.974762 0.974231	0.988986 0.988953 0.988911	0.942684 0.938922 0.932746	0.988996 0.988929 0.988245
4	**800**	0.973324 0.972674 0.972261	0.974733 0.974578 0.974523	0.975472 0.974556 0.974102	0.976521 0.974568 0.973629
6	**1200**	0.972943 0.972556 0.972186	0.972783 0.972692 0.972541	0.973631 0.972647 0.971876	0.974320 0.972655 0.972654
8	**1500**	0.962364 0.961913 0.961347	0.964895 0.964772 0.964431	0.965653 0.960728 0.960022	0.945623 0.944756 0.944032

Table 4.4 Best, average and worst values of HV with N (N>= *Obj Nbr*/2) correlated objectives using 15 independent runs

Obj Nbr	Max Gen	MOEA/D(PBI)	MOEA/DD	NSGA-III	Two_Arch2
3	**400**	0.994887 0.993843 0.983802	0.994233 0.993568 0.993134	0.940232 0.939828 0.939344	0.988764 0.988538 0.988462
4	**800**	0.984426 0.976416 0.976328	0.983652 0.981426 0.976124	0.978863 0.978574 0.976231	0.976498 0.976422 0.976346
6	**1200**	0.971596 0.971574 0.971523	0.977123 0.974581 0.973130	0.978923 0.972402 0.972103	0.974635 0.974582 0.974247
8	**1500**	0.969886 0.969815 0.969702	0.971841 0.969822 0.969723	0.966876 0.966525 0.966234	0.952886 0.952835 0.952803

From the obtained results, it can be concluded that, for different numbers of objectives and generations, MOEA/D is more efficient than the NSGA-III and MOEA/DD is generally more efficient than other algorithms. Moreover, the HV increases if there is a correlation between the different objectives, especially in the case of the MOEA/D algorithm having higher relative advantage improvement compared to other algorithms.

4.2.1.2 Testing the effect of the population size

In this section, HV values are presented for different population sizes in order to test the effect of the variation in the population size on the behavior of MaOAs. The scalarizing function used in MOEA/D and MOEA/DD is PBI (0.5). The probability of mutation is 1/500 (bit-flip mutation) and the probability of recombination is 0.8 (SBX crossover). No neighbor mating of parents and the objectives are correlated (For each experiment with N objectives, at least N/2 of them are correlated). The number of reference points is chosen according to the size of the population and the number of objectives. Table 4.5 details the best, average and worst HV values when varying the population number and the size population.

Table 4.5 Best, average and worst values of HV with different population sizes and various objectives numbers

Obj Nbr	Population size	MOEA/D (PBI)	MOEA/DD	NSGA-III	Two_Arch2	Number of reference points (MOEA/DD, NSGA-III)
4	100	0.956923 0.956517 0.956208	0.983461 0.981027 0.980429	0.973231 0.972675 0.972089	0.973682 0.972987 0.972023	90
	500	0.972863 0.972165 0.972022	0.985237 0.984162 0.984103	0.977863 0.977258 0.977037	0.985682 0.976263 0.976044	130
	1000	0.984426 0.976412 0.976328	0.983652 0.981426 0.976124	0.978863 0.978574 0.976231	0.976498 0.976422 0.976346	255
	1200	0.976664 0.976586 0.972343	0.986213 0.985897 0.985251	0.978683 0.978764 0.978037	0.977023 0.976986 0.976431	280
	1400	0.976874 0.976758	0.986852 0.986238	0.978985 0.978583	0.987875 0.987244	290

		0.974032	0.985140	0.978362	0.977032	
8	100	0.969369 0.969292 0.969083	0.970145 0.969233 0.969002	0.959863 0.959467 0.959302	0.952894 0.952236 0.952035	90
	500	0.969963 0.969643 0.969354	0.969786 0.969645 0.969423	0.960869 0.960098 0.959326	0.969878 0.952543 0.952132	230
	1000	0.969886 0.969815 0.969702	0.971841 0.969822 0.969723	0.966876 0.966525 0.966234	0.952886 0.952835 0.952803	320
	1200	0.969894 0.969831 0.969063	0.970274 0.969962 0.969146	0.967964 0.967663 0.960022	0.970623 0.953195 0.952678	350
	1400	0.969965 0.969887 0.969576	0.970988 0.970231 0.969862	0.968326 0.967989 0.960374	0.971589 0.953651 0.953233	350

For most numbers of objectives, better results were obtained by MOEA/D and MOEA/DD than NSGA-III. Obtained results demonstrate that the increase of the population size does not affect the ability of search of the MOEA/D. Contrariwise; the MOEA/D efficiency is degraded by the rise in the population size and does not work well with large population sizes. As a result, determining the appropriate size of the population according to the number of considered objectives is an interesting area of research. An important observation is that the efficiency of MOEA/D cannot be influenced by the population size increase due to the multiple neighbors which may be replaced with newly-generated better off-spring.

4.2.1.3 Testing the effect of the choice of the scalarizing functions in MOEA/D

Some previous research studies proved that MOEA/D works well on a wide range of test problems despite the fact that its performance depends on the choice of the used scalarizing function. Thus, the choice of the appropriate scalarizing function (or the appropriate scalarizing function set) is an interesting research topic. In our experiments, MOEA/D is applied with the weighted sum, the weighed Tchebycheff and the PBI function (θ = 0,01, 0.5, 1.0, 5.0) to our many-objective problem with α = 1, 1.01, 1.1. The performance of each scalarizing function is evaluated by calculating the average HV value over 15 runs. The population size in MOEA/D is specified as 1000. The mutation probability is 1/500 (Bit-flip mutation) and the recombination probability is 0.8 (SBX crossover). No neighbor mating in recombination and the objectives are either non-correlated or correlated (if correlated, at least N/2 objectives are correlated for each experiment with N objectives). The MOEA/D algorithm is tested using the mentioned scalarizing functions then using the adaptive scalarizing functions concept as discussed previously in the approach (see section 3.3.2.2). Table 4.6 shows the HV values when varying the scalarizing functions with different correlation relations between the different objectives.

Table 4.6 Average values of HV with various scalarizing functions and correlation relations between objectives

		Non-Correlated Objectives				Correlated Objectives			
		2	**4**	**6**	**8**	**2**	**4**	**6**	**8**
Weighted sum		0.969216	0.971227	0.971074	0.970819	0.969812	0.972318	0.972517	0.972603
Weighted Tchebycheff	**α**=1	0.960125	0.959591	0.959117	0.955584	0.975044	0.973581	0.972976	0.970994
	α=1.01	0.960314	0.959783	0.959374	0.955702	0.975210	0.973816	0.973112	0.971253
	α=1.1	0.960722	0.959958	0.959602	0.955958	0.975558	0.974092	0.973376	0.971546
PBI (θ=0.01)	**α**=1	0.969002	0.968797	0.968814	0.968612	0.969973	0.969917	0.969546	0.969411
	α=1.01	0.969105	0.968946	0.968705	0.968517	0.969804	0.969961	0.969472	0.969171
	α=1.1	0.969289	0.968984	0.968462	0.968584	0.969882	0.969983	0.969482	0.969277
PBI (θ=0.5)	**α**=1	0.969458	0.969101	0.969215	0.969547	0.970135	0.970798	0.970814	0.969984
	α=1.01	0.969907	0.969204	0.968827	0.968376	0.970914	0.970813	0.970609	0.969774
	α=1.1	0.969931	0.969286	0.969119	0.968663	0.971221	0.970994	0.970726	0.969631
PBI (θ=1)	**α**=1	0.974432	0.972526	0.972971	0.962134	0.977083	0.976089	0.971938	0.969967
	α=1.01	0.974515	0.972674	0.972556	0.961913	0.977142	0.976416	0.971574	0.969815
	α=1.1	0.974566	0.972738	0.971798	0.961678	0.977361	0.976482	0.971844	0.969892
PBI (θ=5)	**α**=1	0.970682	0.970025	0.968718	0.981823	0.971058	0.971083	0.869988	0.963776
	α=1.01	0.970991	0.970322	0.968203	0.961459	0.971326	0.971184	0.969637	0.963281
	α=1.1	0.970994	0.970537	0.968561	0.961985	0.971632	0.971446	0.969791	0.963688

According to the results in Table 4.6, the weighted Tchebycheff is not appropriate for the many-objective problem with no or small correlation relation between objectives. However, it worked well on our many-objective problem when objectives were correlated. Indeed, the

performance of the weighted Tchebycheff scalarizing function depends on the value of α. If this value is increased from 1 to 1.01 and 1.1, which helps to improve the average HV for all the tested numbers and types (correlated/non-correlated) of objectives.
Concerning the PBI, the parameter α has a variable behavior. For all number of objectives, better results were obtained when these objectives were correlated. Having 2 and 4 objectives, better results (less deterioration) were provided using PBI with α = 1.01 and α = 1.1. However, having 6 and 8 objectives, promising results were obtained using PBI with α =1.0. As far as θ parameter is concerned, good results were provided using the PBI function with θ = 0.01, 0.5 and 1, while the PBI with θ = 5 was always the worst.
Another interesting finding is that, for small penalty (θ) values, the deterioration of the performance of the MOEA/D when increasing the objectives number was less clear when using the PBI function than the weighted Tchebycheff one. Moreover, employing a high penalty values (θ=5) with non-correlated objectives, the MOEA/D encountered difficulties in finding better solutions when using the weighted Tchebycheff function, compared to the PBI.

Besides, the experimental findings show that the weighed sum is not suitable for two-objective problems. However, it is a good choice for four to eight many-objective problems. For further investigations, the evolution of the number of non-dominated solutions according to the size of the population, using different scalarizing functions, can be studied.

4.2.1.4 Testing the effect of using neighborhood mating restrictions and adaptive recombination operators

In this section, we examine the effect of the proposed strategy for mating similar parent and adaptive mutation and recombination operators (see section 3.3.2.1). The performance of each algorithm is evaluated using the average HV over 15 runs. The population size in MOEA/D is specified as 1000. The mutation probability is 1/500 and the recombination probability is 0.8. Neighbor mating is performed in recombination and the objectives are correlated. The number of reference points is set to 100. Table 4.7 illustrates the obtained results.

Table 4.7 Best, average and worst HV values using adaptive operators

	Obj_Nbr	MOEA/D(PBI)	MOEA/DD	NSGA-III	Two_Arch2
Bit-flip mutation / SBX recombination	4	0.984426	0.983652	0.978863	0.976498
		0.976416	0.981426	0.978574	0.976422
		0.976328	0.976124	0.976231	0.976346
	8	0.969886	0.971841	0.966876	0.952886
		0.969815	0.969822	0.966525	0.952835
		0.969702	0.969723	0.966234	0.952803
Using neighborhood mating restrictions and adaptive operators	4	0.978678	0.983129	0.979697	0.977234
		0.978133	0.981952	0.979342	0.976986
		0.976253	0.980237	0.979032	0.976343
	8	0.970489	0.971002	0.971234	0.953864
		0.969932	0.970254	0.967751	0.953366
		0.963231	0.968968	0.963268	0.953032

The obtained results with different numbers of objectives indicate that the neighborhood mating and the use of adaptive operators improve considerably the search performance. In fact, better results were obtained for different numbers of objectives on the MOEA/DD algorithm. Obviously, when the number of objectives increases, the advantage of the MOEA/DD over the MOEA/D becomes clearer. Nevertheless, MOEA/D improves more considerably the average HV value (with and without similar parent recombination), compared to other algorithms. Moreover, experimental results show that mating similar parents improves the diversity without deteriorating the convergence.

4.2.1.5 Testing the effect of hybridizing the 5EMOs with a dimensionality reduction approach

In this section, we investigate the effect of incorporating our proposed approach for dimensionality reduction. In this set of experiments, the HV is calculated and the size of the population is set to 1000. Different adaptive scalarizing functions are used for MOEA/D and

MOEA/DD. The mutation probability is set to 1/400 with an index of 50 and the recombination probability is 0.9 with an index of 5. Adaptive mutation and recombination operators are used with neighbor parents mating. 8 correlated objectives are employed (for each experiment with N objectives, at least N/2 objectives are correlated). 250 reference points are applied for NSGA-III. Table 4.8 illustrates the obtained results.

Table 4.8 Best, average and worst HV values obtained before and after applying the dimensionality reduction approach

	Initial Obj Nbr	Obj Nbr after reduction	MOEA/D (PBI)	MOEA/DD	NSGA-III	Two_Arch2
Without reduction	4 / 5	4 / 5	0.984426 0.976416 0.976328	0.983652 0.981426 0.976124	0.978863 0.978574 0.976231	0.976498 0.976422 0.976346
Using L- PCA /NL-MVU-PCA		3	0.994975 0.991264 0.980023	0.982897 0.982542 0.982231	0.982896 0.982251 0.982033	0.989352 0.987144 0.986021
Without reduction	8	8	0.969886 0.969815 0.969702	0.971841 0.969822 0.969723	0.966876 0.966525 0.966234	0.952886 0.952835 0.952803
Using L- PCA /NL-MVU-PCA		4	0.971978 0.970951 0.970236	0.984986 0.984158 0.983943	0.969897 0.961152 0.960364	0.954237 0.953654 0.953028

From the results presented in Table 4.8, for four and eight objectives, the HV values found when using dimensionality reduction approach are higher than those obtained without using this method due to the reduction in the number of objectives from eight to five in the case of our real-world problem. Moreover, the improvement rate of the MOEA/D clearly exceeds those of other algorithms.

In the following, we detail the application of the proposed dimensionality reduction approach (The NL-MVU-PCA algorithm, for non-linear objective reduction, and L-PCA for linear objective reduction) to our real-world problem. Since Two_Arch2 has the best performance, we describe here only the use of the MOEA/DD as a MaOA to test our dimensionality reduction approach on an eight-objective 3D deployment problem. The tables below illustrate the set of the most dominant objectives found after 15-runs of the NL-MVU-PCA and L-PCA. According to (Sinha et al. 2013), starting from an initial set of objectives $F_0=\{f_1,\ldots, f_M\}$, the NL-MVU-PCA aims at identifying the set F_T of essential objectives by applying the following steps:

Step 1 (Computing the correlation and the kernel matrix): Based on the input data, the correlation matrix R (for linear objective reduction (L-PCA)) is plotted, the kernel matrix K (for nonlinear objective reduction (NL-MVU-PCA)) and its principal component (eigenvectors and eigenvalues) are computed. According to (Sinha et al. 2013), $R=(1/M).XX^T$ where M is the number of objectives and X is the input data. K is also calculated according to the formulation in (Sinha et al. 2013). Tables 4.9, 4.10, 4.11 and 4.12 illustrate the values of the matrix R, K as well as their Eigenvectors and eigenvalues.

Table 4.9 The correlation Matrix R on the first iteration

	f1	f2	f3	f4	f5	f6	f7	f8
f1	1	-0.458	0.896	0.985	-0.258	0.885	0.875	-0.647
f2	-0.458	1	-0.521	-0.678	-0.735	-0.613	0.365	-0.647
f3	0.896	-0.521	1	0.982	-0.385	0.997	0.354	-0.647
f4	0.985	-0.678	0.982	1	-0.392	0.839	0.365	-0.647
f5	-0.258	-0.735	-0.385	-0.392	1	-0.264	0.238	-0.647
f6	0.885	-0.613	0.997	0.839	-0.264	1	0.364	-0.647
f7	0.875	0.365	0.354	0.365	0.238	0.364	1	-0.647
f8	-0.647	-0.647	-0.647	-0.647	-0.647	-0.647	-0.647	1

Table 4.10 The kernel Matrix K

	f1	*f2*	*f3*	*f4*	*f5*	*f6*	*f7*	*f8*
f1	4.568	-4.521	4.895	6.552	-7.353	9.652	6.548	-6.365
f2	-4.521	6.021	-4.257	-6.215	-7.985	-9.245	9.253	-6.365
f3	4.895	-4.257	6.892	6.812	-7.154	11.246	12.568	-6.365
f4	6.552	-6.215	6.812	8.453	-7.554	11.246	12.568	-8.852
f5	-7.353	-7.985	-7.154	-7.554	12.258	-11.246	18.258	-8.254
f6	9.652	-9.245	11.246	11.246	-11.246	18.254	18.891	-8.852
f7	6.548	9.253	12.568	12.568	18.258	18.891	25.547	-18.255
f8	-6.365	-6.365	-6.365	-8.852	-8.254	-8.852	-18.255	21.541

Table 4.11 Eigenvectors and eigenvalues of the matrix R

e1=0.646 v1	e2=0.221 v2	e3=0.084 v3	e4=0.003 v4
0.215	0.886	0.568	0.638
-0.568	-0.322	0.546	-0.457
0.585	0.662	0.531	-0.891
0.689	-0.211	0.284	-0.457
-0.985	-0.354	0.893	0.594
0.325	-0.498	0.045	0.617
0.236	-0.158	0.104	0.685
-0.652	-0.659	-0.593	-0.237

Table 4.12 Eigenvectors and eigenvalues of the matrix K

e1=0.548 v1	e2=0.276 v2	e3=0.048 v3	e4=0.002 v4
-0.234	0.056	0.448	-0.253
0.665	-0.094	0.125	0.151
0.652	0.114	0.356	0.198
0.745	0.146	0.651	0.235
0.351	-0.338	0.821	0.358
0.452	0.562	0.886	0.564
-0.635	0.567	-0.662	0.282
0.328	-0.523	-0.543	0.025

Step 2 (Eigenvalue Analysis): consists in identifying the set of the important objectives in the initial set of objectives by performing the eigenvalue analysis that identifies the principal components (directions of significant variance) in the data. Table 4.13 (Table 4.14, respectively) depicts the eigenvalue analysis for linear objective reduction (non-linear objective reduction, respectively).

Table 4.13 Eigenvalue Analysis for L-PCA

PCA (N°)	Variance (%)	Cumulative (%)	Selected objectives							
1	64.6	64.60		f2			f5	f6	f7	f8
2	22.1	97.09	f1	f2	f3	f4	f5	f6	f7	f8
3	8.4	99.62		f2		f4		f6	f7	f8
4	0.3	99.99		f2			f5		f7	f8

Table 4.14 Eigenvalue Analysis for NL-MVU-PCA

PCA (N°)	Variance (%)	Cumulative (%)	Selected objectives							
1	54.8	54.80	f1	f2	f3		f5	f6		
2	27.6	94.45	f1			f4	f5		f7	f8
3	4.8	99.99		f2			f5		f7	f8
4	0.2	99.99		f2	f3	f4			f7	f8

Step 3 (Reduced Correlation Matrix Analysis): consists in identifying the set of identically-correlated subsets by carrying out the reduced correlation matrix analysis. The important objectives in each subset are retained and other objectives are discarded, which allows further reduction of the objective set obtained after step 2. Table 4.15 (Table 4.16, respectively) represents the RCM analysis for linear objective reduction (non-linear objective reduction, respectively). According to (Saxena et al. 2013), $T_{cor} = 1.0 - e_1.(1.0 - M'/M)$ where M' denotes the number of principal components required to account for 95.4% variance and M is the number of the problem objectives.

Table 4.15 RCM analysis for L-PCA

Potential identically correlated set(s)	{f1,f3,f4,f6}
T_{cor} (correlation threshold)	1.0-0.646(2/8)=0.8385
Identically correlated set(s)	{f1,f3,f4,f6}

Table 4.16 RCM analysis for NL-MVU-PCA

Potential identically correlated set(s)	{f1,f3,f4,f6}
T_{cor} (correlation threshold)	1.0-0.548(3/8)=0.7945
Identically correlated set(s)	{f1,f3,f4,f6}

Step 4 (Selection scheme): consists in identifying the most important objective in each set by applying the selection scheme. Table 4.17 (Table 4.18, respectively) demonstrates the selection scheme for linear objective reduction (non-linear objective reduction, respectively).

Table 4.17 Selection scheme for L-PCA

	e_1= 0.646 V_1	e_2= 0.221 V_2	e_3= 0.084 V_3	e4=0.003 v4	Objective selection Score
f1	0.215	0.886	0.568	0.638	0.458
f1	-0.568	-0.322	0.546	-0.457	0.462
f1	0.585	0.662	0.531	-0.891	0.483
f1	0.689	-0.211	0.284	-0.457	0.494

Table 4.18 Selection scheme for NL-MVU-PCA

	e_1= 0.548 V_1	e_2= 0.276 V_2	e_3= 0.048 V_3	e4=0.002 v4	Objective selection Score
f1	-0.234	0.056	0.448	-0.253	0.238
f1	0.665	-0.094	0.125	0.151	0.295
f1	0.652	0.114	0.356	0.198	0.684
f1	0.745	0.146	0.651	0.235	0.793

Step 5 (Computation of the error): is to measure the error incurred in one iteration of the proposed framework. This measure calculates the unaccounted left variance when discarding the objectives constituting the redundant objective set. According to the equation proposed in (Saxena et al. 2013), the error for L-PCA (NL-MVU-PCA, respectively) is equal to 0.000239 (0.000458, respectively).

The above-mentioned five-step process is achieved iteratively until the set of necessary objectives will be reduced to two objectives or until it stills the same for two successive iterations.

4.2.1.6 Testing the effect of hybridizing the EMOs with dimensionality reduction and user preferences

In this section, we examine the effect of applying our proposed approach to incorporate both dimensionality reduction method and user preference one. In this set of experiments, the HV is calculated with a large population (1000). The used scalarizing function (for MOEA/D and MOEA/DD) is PBI (0.5). The mutation probability is 1/400 with an index of 50, and the recombination probability is 0.9 with an index of 5. Adaptive mutation and recombination operators are employed with neighbor parents mating. 8 correlated objectives are used (for each experiment with N objectives, at least N/2 objectives are correlated). We use 250 reference points for NSGA-III. After applying reduction approach, the preference is applied on a reduced set of objectives. Tables 4.19 and 4.20 show the final solutions specifications (using Two_Arch2 as an EMO). Each run has a different initial population, which is the result of applying our reduction procedure on the concerned EMO. d_s is a user-defined parameter representing the expected improvement in solutions obtained from the current best solution based on the value function, and d_s =0.01. TD_{Max} =30 is the maximum number of calls of the preference information introduced by the DM.

Table 4.19 Median obtained solutions (objective values)

		The most preferred point satisfying the KKT conditions and used to construct the value function.	Average values			
			MOEA/D	NSGA-III	Two_Arch2	MOEA/DD
f1	Number of added nomad nodes	128.452	134.161	142.54	152.339	133.581
f2	Energy consumption	3.857	3.998	4.021	4.056	3.962
f3 (redundant)	Hardware deployment cost	85	88.468	96.184	93.923	88.646
f4 (redundant)	Network Utilization	1.00	0.946	0.796	0.849	0.962
f5	Localization rate	3.991	3.605	3.882	3.863	3.812
f6 (redundant)	Coverage rate	5.865	4.189	4.984	4.235	5.572
f7	Lifetime	4280	3885	3687	3956	4065
f8	Connectivity rate	189.89	168.524	166.515	168.542	174.266

Table 4.20 Median distance of the solutions obtained from the most preferred solutions

	MOEA/D	NSGA-III	Two_Arch2	MOEA/DD
Accuracy	0.234	0.419	0.468	0.023
Number of function evaluations	6321	7945	8231	5895
Number of required DM calls	TD_{Max}	26	25	16

4.2.1.7 Results on PI-NSGA-III-VF

In this section, the algorithms parameters and the obtained numerical results are presented. To assess the quality of the results, several metrics can be used. The Hypervolume (HV) (Zitzler et al. 1999) is one of these metrics. Although its computational cost is high, it is ideal in our case (real world problem) because of the unknown true PF. The used parameters setting are detailed in Table 4.21. The platform PlatEMO (Tian et al. 2017) is used to implement NSGA-III and PI-EMO-VF algorithms.

Table 4.21 Parameters setting of the algorithms

Parameter		Value
Population size		300
Recombination	Operator	SBX
	probability	0.8
	distribution index	45
Mutation	Operator	bit-flip
	probability	1/400
	distribution index	25
Number of independent runs		25, on different initial populations
Number of constraints		7
Number of objectives		Variable, see Table 4.22
Maximum number of generations		Variable, see Table 4.22
maximum number of preference calls		30
Improvement in solutions (d_s)		0.01

Given the random behavior of the optimization algorithms and aiming at obtaining statistically reliable comparison results, an average of 25 executions of the algorithms is performed. Table 4.22 demonstrates the obtained HV values for different number of objectives and number of generations. Best performances are shown with shaded backgrounds.

Table 4.22 Hypervolume values (Best, average and worst)

Number of Objectives	Max nbr of generations	MOEA/DD	PI-NSGA-III-VF
3	1300	0.902231 0.901658 0.898235	0.902458 0.901896 0.898023
4	1800	0.974892 0.974743 0.973897	0.974685 0.974233 0.973612
5	2600	0.972983 0.972563 0.972826	0.972892 0.972716 0.972684

In our real world context, the PI-NSGA-III-VF performs better when the number of objectives exceeds three. This matches the ascertainment of the NSGA-III authors which indicates that the NSGA-III is more appropriate for resolving many-objectives problems. Moreover, better HV values are retained with smaller number of objectives; this is explained by the proportional dependence between the problem complexity and the number of objectives.
Tables 4.23 and 4.24 show the final solutions specifications, for $\boldsymbol{d_s}$ =0.01, after applying the preference.

Table 4.23 Average retained objective values

Objective	Mpp*	Average-values	
		MOEA/DD	PI-NSGA-III-VF
f4 (Quality of links)	91	85.985	92.223
f2 (Utilization of network)	1.000	0.947	0.786
f5 (Rate of localization)	3.992	3.602	3.847
f3 (Rate of coverage)	4.986	4.194	4.853
f1 (Rate of connectivity)	190.24	168.633	166.612

Mpp^* is the most preferred point satisfying the KKT conditions and used to construct the value function.

Table 4.24 Average distance between the most preferred solutions and the obtained ones

	MOEA/DD	PI-NSGA-III-VF
Accuracy	0.247	0.426
Number of function evaluations	6343	7948
Number of required DM calls	23	25

4.3 Numerical results on unconstrained DTLZ test functions

After testing our approach on real testbeds, we present the results provided by using a set of benchmark test problems to prove its scalability. Using 15 independent runs in each test, the performances of MOEA/D (Zhang and Li 2007) and those of MOEA/DD (Li et al. 2015), NSGA-III (Deb and Jain 2014) and Two_Arch2 (Wang et al. 2015) are verified. The experiments are conducted on a set of test problems taken from the widely-used test suite DTLZ (Deb et al. 2005). For each test problem, 3, 5, 10 and 15 objectives are considered, respectively. To compare the different algorithms in terms of the quality of the obtained non-dominated solution sets, the IGD indicator is used. As explained above, IGD is a metric which allows measuring both the diversity and the convergence of the obtained non-dominated solutions. It corresponds to the average Euclidean distance separating each reference solution from its closest non-dominated one. As reference solutions, the set of Pareto optimal solutions generated by all algorithms over all runs are used.

The applied test functions are DTLZ1 to DTLZ4 (Deb et al. 2005) and DTLZ(I,M) (Deb and Saxena 2006) from the DTLZ test suite. The main advantages of the DTLZ test problem suite are its scalability and its ability to arbitrary specifying the number of objectives. Indeed, for all DTLZ problems, the number of objectives varies from 3 to 15, i.e., $m \in \{3, 5, 8, 10, 15\}$. For instance, DTLZ1-4 are non-constrained and scalable three- to fifteen-objective problems. According to the recommendations in (Deb et al. 2005), the number of decision variables for DTLZ test instances is set to $n = m + r - 1$, where $r = 5$ for DTLZ1 and $r = 10$ for DTLZ2, DTLZ3 and DTLZ4. Table 4.25 illustrates each test problem features.

Table 4.25 Features of the used DTLZ test problems

Used test problem	Features	Reducibility of the problem (Redundancy of objectives)
DTLZ1	Scalable, Linear, Multi-modal	Irreducible problem (conflicting objectives)
DTLZ2	Scalable and continuous many-objective minimization problem, Concave Pareto Front	Irreducible problem (conflicting objectives)
DTLZ3	Scalable, Concave, Multi-modal	Irreducible problem (conflicting objectives)
DTLZ4	Scalable, Concave, Biased, favors the diversity of solutions	Irreducible problem (conflicting objectives)
DTLZ5(I,M)	Scalable problem, Non-convex Pareto-optimal front	Reducible problem (since it has M − I + 1 redundant objectives But if I=M-1, then it becomes an irreducible problem)

In our experiments, the DTLZ5 (I, M) test problem suit is also used. In fact, DTLZ5 (*I, M*) is an improvement of the DTLZ5 test problem characterized by a degraded Pareto front, which represents a good feature for dimensionality reduction. However, non-redundant objectives in DTLZ5 have a fixed number, which does not allow testing properly the efficiency of the algorithms. As a solution, the DTLZ5(*I, M*) test problem suit was proposed by Deb and Saxena (2006). The DTLZ5 (*I, M*) allows changing, without influencing to each other, the objectives number in the problem (noted *M*), the non-dominated objectives number (the dimensionality of the Pareto-optimal surface, noted *I*) and the variables number (noted *n*). Moreover, this test problem allows changing the dimensionality (I) of the Pareto-optimal front by simply setting I to an integer between two and M. The first M − I + 1 objectives are correlated, and the rest are in conflict with the other objectives. For example, having I = 2, a minimum of two objectives (the objective f_M and any other one) is sufficient to represent the correct Pareto-optimal front as it is the case when DTLZ5(2,10) problem has 10 objectives and two-dimensional Pareto-optimal front. Eight of these objectives are redundant and the Pareto-optimal front can be found, for different scales, with the objective f10 and any other objective function. Another instance of the DTLZ(I,M) problem, is the DTLZ5(3,10) involving 10 objectives and having a three-dimensional Pareto-optimal front.

To sum up, in order to investigate the degree of dependence between objectives, we test the propose algorithms on a highly-redundant problem (DTLZ5(2,15)), a moderately-redundant problem (DTLZ5(3,5)) and a non-redundant problem (DTLZ1 to DTLZ4).
Unless it is differently indicated in each experiment, the objectives are correlated (For each experiment with N objectives, at least N/2 of the objectives are correlated).

4.3.1 Testing the effect of interdependence between objectives

In this section, we study the effect of having dependencies between objectives. The size of the population is set to 1000 (a large population), which is run for different number of generations. The used scalarizing function (for MOEA/D and MOEA/DD) is PBI (5.0). The mutation probability is 1/400 with an index of 50 (Bit-flip mutation). The recombination probability is 0.9 with an index of 5 (SBX crossover). There are no neighbor mating and adaptive recombination in this set of experiments. The objectives are correlated (for each experiment with N objectives, at least N/2 objectives are correlated). 250 reference points for NSGA-III, with 15 independent runs, are used. Table 4.26 illustrates the median values of IGD on irreducible test functions (DTLZ1-2). Table 4.27 represents the best, average and worst values of IGD on reducible test functions (DTLZ5(2,15) and DTLZ5(3,5)). The best performance for each instance is shown with a gray background.

Table 4.26 Best, average and worst IGD values on DTLZ1-4 problems

Test Pb	Obj Nbr	Number of generations	MOEA/D (PBI)	MOEA/DD	NSGA-III	Two_Arch2
DTLZ1	3	400	4.095E-4 1.495E-3 4.743E-3	3.191 E-4 5.848 E-4 6.573 E-4	4.880 E-4 1.308 E-3 4.880 E-3	1.089 E-3 1.228 E-3 1.523 E-3
	5	600	3.179E-4 6.372E-4 1.635E-3	2.635 E-4 2.916 E-4 3.109 E-4	5.116 E-4 0.980 E-4 1.979 E-3	6.034 E-4 6.372 E-4 6.967 E-4
	10	1000	3.872E-3 5.073E-3 6.132E-3	1.828 E-3 2.225 E-3 2.467 E-3	2.215 E-3 3.462 E-3 6.869 E-3	2.368 E-3 2.962 E-3 3.101 E-3
	15	1500	1.236E-2 1.431 E-2 1.692 E-2	2.867 E-3 4.203 E-3 4.609 E-3	2.649 E-3 5.063 E-3 1.123 E-2	4.346 E-3 4.927 E-3 5.239 E-3
DTLZ2	3	250	5.432 E-4 6.406 E-4 8.006 E-4	6.666 E-4 8.073 E-4 1.243 E-3	1.262 E-3 1.357 E-3 2.114 E-3	6.884 E-4 7.122 E-4 7.697 E-4
	5	350	1.219 E-3 1.437 E-3 1.727 E-3	1.128 E-3 1.291 E-3 1.424 E-3	4.254 E-3 4.982 E-3 5.862 E-3	4.203 E-4 4.760 E-4 4.992 E-4
	10	750	2.474 E-3 2.778 E-3 3.235 E-3	3.223 E-3 3.752 E-3 4.145 E-3	1.350 E-2 1.528 E-2 1.697 E-2	1.104 E-2 1.459 E-2 1.858 E-2
	15	1000	5.254 E-3 6.005 E-3 9.409 E-3	4.577 E-3 5.863 E-3 6.929 E-3	1.360 E-2 1.726 E-2 2.114 E-2	4.263 E-3 4.726 E-3 5.294 E-3
DTLZ3	3	1000	9.773 E-4 3.426 E-3 9.113 E-3	5.690 E-4 1.892 E-3 6.231 E-3	9.751 E-4 4.007 E-3 6.665 E-3	3.237 E-3 3.742 E-3 4.123 E-3
	5	1000	1.129 E-3 2.213 E-3 6.147 E-3	6.181 E-4 1.181 E-3 4.736 E-3	3.086 E-3 5.960 E-3 1.196 E-2	4.364 E-3 4.834 E-3 6.230 E-3
	10	1500	2.791 E-3 4.319 E-3 1.010 E+0	1.689 E-3 2.164 E-3 3.226 E-3	8.849 E-3 1.188 E-2 2.082 E-2	9.265 E-3 9.836 E-3 2.364 E-2
	15	2000	4.360 E-3 1.664 E-2 1.260E+0	5.716 E-3 7.461 E-3 1.138 E-2	1.401 E-2 2.145 E-2 4.195 E-2	1.632 E-2 2.068 E-2 2.894 E-2
DTLZ4	3	600	2.929 E-1 4.280 E-1 5.234 E-1	1.025 E-4 1.429 E-4 1.881 E-4	2.915 E-4 5.970 E-4 4.286 E-1	5.236 E-4 5.784 E-4 7.982 E-4
	5	1000	1.080 E-1 5.787 E-1 7.348 E-1	1.097 E-4 1.296 E-4 1.532 E-4	9.849 E-4 1.255 E-3 1.721 E-3	1.032 E-4 1.225 E-4 1.687 E-4
	10	2000	3.966 E-1 9.203 E-1 1.077E+0	1.291 E-3 1.615 E-3 1.931 E-3	5.694 E-3 6.337 E-3 1.067 E-1	0.896 E-3 1.187 E-3 1.763 E-3
	15	3000	5.890 E-1 1.133 E+0 1.249 E+0	1.474 E-3 1.881 E-3 3.159 E-3	7.110 E-3 3.431 E-1 1.073 E+0	3.024 E-1 3.268 E-1 3.903 E-1

Table 4.27 Best, average and worst IGD values on DTLZ5(2,15) and DTLZ5(3,5) problems

Test Problem	MOEA/D (PBI)	MOEA/DD	NSGA-III	Two_Arch2
DTLZ5(2,15)	1.134 E-2	1.641 E-3	2.034 E-3	1.344 E-3
	1.326 E-2	3.195 E-3	4.985 E-3	2.786 E-3
	1.667 E-2	4.034 E-3	9.031 E-3	4.239 E-3
DTLZ5(3,5)	1.194 E-4	0.634 E-4	1.985 E-4	0.736E-4
	1.265 E-4	3.056 E-4	3.192 E-4	0.925E-4
	1.703 E-4	6.238 E-4	8.054 E-4	2.362E-4

In the majority of cases, for different number of objectives and number of generations, MOEA/DD has the minimum (best) IGD value. Generally, MOEA/DD is more performing than NSGA-III. The best amount of improvement in IGD values between correlated and non-correlated objectives is noted for the MOEA/D algorithm. As a consequence, we can conclude that MOEA/D works well on the many-objective case with highly-correlated objectives, which indicates clearly the advantage of using dependent objectives in many-objective problems. This finding proves that algorithms which are not designed for MaOPs (such as MOEA/D) may be efficient in resolving MaOPs if the objectives are correlated. Compared with the IGD values using non-correlated objectives (not presented here due to space constraints), the obtained results illustrated in Table 4.26, show that the IGD values decrease if the objectives are correlated.

4.3.2 Testing the effect of the size of the population

In this section, we show the effect of varying the size of the population on the quality of the found solutions. For each test problem, different specifications of the population size in each MOEA are used: 100 (small), 600, 1000 1200 and 1400 (large). PBI (5.0) is employed as a scalarizing function for MOEA/D and MOEA/DD. The mutation probability is 1/400 with an index of 50 (Bit-flip mutation), and the recombination probability is 0.9 with an index of 5 (SBX crossover). No neighbor mating in recombination. We apply 250 reference points for NSGA-III algorithms. Table 4.28 (Table 4.29, respectively) illustrates the results of measuring the IGD on the DTLZ1 and DTLZ2 (DTLZ5(2,15) and DTLZ5(3,5), respectively) test functions using the above-indicated parameters.

Table 4.28 Best, average and worst IGD values on DTLZ1 and DTLZ2 for different population sizes

Test Pb	Obj Nbr	Population size	MOEA/D (PBI)	MOEA/DD	NSGA-III	Two_Arch2	Number of reference points
DTLZ1	5	100	3.031 E-4	1.036 E-4	0.679 E-3	2.364 E-4	90
			6.627 E-4	4.145 E-4	1.325 E-3	6.569 E-4	
			9.309 E-4	8.327 E-4	1.325 E-3	1.412 E-4	
		600	6.023 E-4	2.264 E-4	4.356 E-4	2.034 E-4	130
			6.466 E-4	3.058 E-4	0.996 E-3	6.433 E-4	
			6.924 E-4	4.065 E-3	7.365 E-3	9.857 E-4	
		1000	3.179 E-4	2.635 E-4	5.116 E-4	6.034 E-4	255
			6.372 E-4	2.916 E-4	0.980 E-4	6.372 E-4	
			1.635 E-3	3.109 E-4	1.979 E-3	6.967 E-4	
		1200	1.872 E-4	7.326 E-4	7.034 E-4	3.302 E-4	280
			6.334 E-4	2.874 E-4	0.976 E-3	6.313 E-4	
			9.659 E-4	0.023 E-4	4.321 E-3	8.034 E-4	
		1400	2.654 E-4	0.791 E-4	4.032 E-4	9.623 E-4	290
			6.136 E-4	2.823 E-4	0.963 E-3	6.284 E-4	
			7.327 E-4	5.475 E-4	3.027 E-3	9.371 E-4	
	10	100	5.084 E-3	2.261 E-3	3.432 E-3	2.674 E-3	270
			5.163 E-3	2.624 E-3	3.863 E-3	2.897 E-3	
			5.369 E-3	2.983 E-3	3.909 E-3	2.978 E-3	
		600	5.032 E-3	2.169 E-3	3.308 E-3	2.636 E-3	290
			5.125 E-3	2.231 E-3	3.734 E-3	2.881 E-3	
			5.421 E-3	2.564 E-3	3.907 E-3	2.908 E-3	
		1000	3.872 E-3	1.828 E-3	2.215 E-3	2.368 E-3	320
			5.073 E-3	2.225 E-3	3.462 E-3	2.962 E-3	
			6.132 E-3	2.467 E-3	6.869 E-3	3.101 E-3	
		1200	4.908 E-3	2.022 E-3	3.123 E-3	2.137 E-3	350
			5.032 E-3	2.195 E-3	3.416 E-3	2.302 E-3	
			5.309 E-3	2.607 E-3	3.709 E-3	2.864 E-3	
		1400	4.364 E-3	2.026 E-3	3.266 E-3	2.023 E-3	350
			4.936 E-3	2.137 E-3	3.403 E-3	2.208 E-3	
			5.211 E-3	2.734 E-3	3.708 E-3	2.604 E-3	
DTLZ2	5	250	1.302 E-3	1.239 E-3	4.364 E-3	4.236 E-4	90
			1.603 E-3	1.698 E-3	4.992 E-3	4.512 E-4	
			1.964 E-3	1.804 E-3	5.238 E-3	4.704 E-4	
		350	1.103 E-3	1.448 E-3	4.337 E-3	3.705 E-4	130
			1.548 E-3	1.675 E-3	4.994 E-3	4.544 E-4	
			1.635 E-3	1.906 E-3	5.208 E-3	5.509 E-4	
			1.303 E-3	1.307 E-3	4.108 E-3	3.307 E-4	255

		750	1.504 E-3 1.607 E-3	1.489 E-3 1.603 E-3	4.987 E-3 5.230 E-3	4.734 E-4 6.228 E-4	
		1000	1.228 E-3 1.437 E-3 1.727 E-3	1.128 E-3 1.291 E-3 1.424 E-3	4.254 E-3 4.982 E-3 5.862 E-3	4.203 E-4 4.760 E-4 4.992 E-4	**280**
		1200	1.130 E-3 1.413 E-3 1.603 E-3	1.195 E-3 1.224 E-3 1.363 E-3	4.109 E-3 4.982 E-3 5.307 E-3	3.438 E-4 4.768 E-4 7.126 E-4	**290**
	10	**250**	2.423 E-3 2.681 E-3 3.967 E-3	3.367 E-3 3.862 E-3 4.063 E-3	1.408 E-2 1.679 E-2 1.838 E-2	1.206 E-2 1.624 E-2 2.368 E-2	**270**
		350	2.490 E-3 2.692 E-3 3.309 E-3	3.404 E-3 3.833 E-3 4.068 E-3	1.408 E-2 1.633 E-2 1.904 E-2	1.326 E-3 1.597 E-2 2.085 E-2	**290**
		750	2.235 E-3 2.724 E-3 3.058 E-3	3.368 E-3 3.794 E-3 4.234 E-3	1.209 E-2 1.578 E-2 1.802 E-2	1.206 E-3 1.534 E-2 2.668 E-2	**320**
		1000	2.474 E-3 2.778 E-3 3.235 E-3	3.223 E-3 3.752 E-3 4.145 E-3	1.350 E-2 1.528 E-2 1.697 E-2	1.104 E-2 1.459 E-2 1.858 E-2	**350**
		1200	2.308 E-3 2.703 E-3 3.024 E-3	3.367 E-3 3.604 E-3 4.089 E-3	1.403 E-2 1.504 E-2 1.868 E-2	1.236 E-3 1.423 E-2 2.634 E-2	**350**

Table 4.29 Best, average and worst IGD values on DTLZ(2,15) et DTLZ(3,5) for different population sizes

Test Pb	**Population size**	**MOEA/D (PBI)**	**MOEA/DD**	**NSGA-III**	**Two_Arch2**	**Number of reference points (MOEA/D, NSGA-III)**
DTLZ5(2,15)	**100**	7.014 E-2 8.032 E-2 0.026 E-1	2.785 E-2 3.034 E-2 6.327 E-2	1.923 E-3 7.985 E-2 8.231 E-2	2.428 E-2 3.024 E-2 6.320 E-2	**90**
	600	2.453 E-2 2.978 E-2 3.127 E-2	4.985 E-3 7.088 E-3 8.124 E-3	2.129 E-3 4.093 E-2 7.055 E-2	3.216 E-3 9.601 E-3 0.995 E-2	**130**
	1000	1.134 E-2 1.326 E-2 1.667 E-2	1.641 E-3 3.195 E-3 4.034 E-3	2.034 E-3 4.985 E-3 9.031 E-3	1.344 E-3 2.786 E-3 4.239 E-3	**255**
	1200	1.012 E-2 1.023 E-2 1.985 E-2	2.385 E-3 2.632 E-3 4.158 E-3	2.023 E-3 4.981 E-3 5.302 E-3	2.029 E-3 2.634 E-3 5.870 E-3	**280**
DTLZ5(3,5)	**350**	0.931 E-4 1.498 E-4 4.682 E-4	2.805 E-4 3.123 E-4 5.301 E-4	3.137 E-4 3.203 E-4 5.112 E-4	1.896 E-4 4.149 E-4 4.268 E-4	**130**
	750	1.235 E-4 1.402 E-4 1.638 E-4	2.083 E-4 3.069 E-4 5.360 E-4	2.043 E-4 3.198 E-4 6.302 E-4	0.932 E-4 1.145 E-4 6.029 E-4	**255**
	1000	1.194 E-4 1.265 E-4 1.703 E-4	0.634 E-4 3.056 E-4 6.238 E-4	1.985 E-4 3.192 E-4 8.054 E-4	0.736E-4 0.925E-4 2.362E-4	**280**
	1200	1.134 E-4 1.203 E-4 4.678 E-4	1.869 E-4 2.863 E-4 4.864 E-4	2.793 E-4 3.107 E-4 5.027 E-4	0.760 E-4 0.887 E-4 4.561 E-4	**290**

Results presented in Table 4.28 show that MOEA/DD has the best IGD median values in DTLZ1 test function for different population sizes and different objective numbers. However, in DTLZ2, MOEA/D and Two_Arch2 share the best IGD median values. Another interesting finding, in this set of experiments, is that the tested algorithms do not have the same behavior as the previously-tested real world problem. For example, MOEA/D variants are better than NSGA-III in DTLZ1. Moreover, it can be concluded from the experimental results that determining an appropriate population size according to the number of objectives is an important area of research.

4.3.3 Testing the effect of the choice of the scalarizing functions in MOEA/D

In this section we test the effect of the choice of the scalarizing functions in MOEA/D for different population sizes and objective numbers. In our experiments we use the following scalarizing functions: the weighed sum, the weighted Tchebycheff with $\alpha = 1, 1.01, 1.1$, and the PBI with $\theta = 0.01, 0.5, 1, 5$ for $\alpha = 1, 1.01, 1.1$. The median IGD values are calculated for each scalarizing function over 15 runs. The objective number varies from 3 to 15. The population is set to 1000 and the number of generations varies from 400 to 1500 for DTLZ1 and from 250 to 1000 for DTLZ2. The mutation probability is 1/500 (Bit-flip mutation) and the recombination probability is 0.8 (SBX crossover). There is no neighbor mating in recombination. The results are shown in Table 4.30.

Table 4.30 IGD median values on DTLZ1 and DTLZ2 for different population sizes

Test problem		DTLZ1				DTLZ2			
Objective number		3	5	10	15	3	5	10	15
Number of generations		400	600	1000	1500	250	350	750	1000
Weighted sum		1.361 E-4	5.642 E-4	6.208 E-3	2.763 E-2	1.210 E-4	1.897 E-3	2.236 E-3	3.864 E-3
Weighted Tchebycheff	α=1	6.641 E-3	3.618 E-4	0.905 E-3	3.087 E-2	2.431 E-4	4.426 E-4	8.631 E-4	2.632 E-3
	α=1.01	2.103 E-3	3.172 E-4	0.686 E-3	3.005 E-2	2.273 E-4	1.267 E-4	7.087 E-4	1.639 E-3
	α=1.1	0.932 E-4	2.484 E-4	0.427 E-3	2.945 E-2	1.163 E-4	9.105 E-5	6.043 E-4	0.984 E-3
PBI (θ=0.01)	α=1	1.132 E-4	6.450 E-4	4.874 E-3	4.032 E-3	8.464 E-4	2.234 E-3	3.461 E-3	6.821 E-3
	α=1.01	2.532 E-3	7.896 E-4	6.890 E-3	5.281 E-2	8.791 E-4	2.467 E-3	5.032 E-3	8.058 E-3
	α=1.1	1.986 E-4	5.783 E-4	4.462 E-3	4.230 E-3	8.469 E-4	2.391 E-3	3.467 E-3	7.364 E-3
PBI (θ=0.5)	α=1	1.925 E-4	6.652 E-4	4.687 E-3	3.975 E-3	8.269 E-4	1.618 E-3	3.102 E-3	5.983 E-3
	α=1.01	2.452 E-3	7.139 E-4	6.302 E-3	4.659 E-2	8.135 E-4	2.109 E-3	4.689 E-3	7.364 E-3
	α=1.1	2.238 E-4	6.754 E-4	4.892 E-3	3.624 E-3	7.149 E-4	2.462 E-3	3.236 E-3	6.203 E-3
PBI (θ=1)	α=1	0.923 E-4	5.036 E-4	2.367 E-3	2.231 E-3	8.146 E-4	1.539 E-3	2.892 E-3	5.682 E-3
	α=1.01	2.102 E-3	6.568 E-4	5.236 E-3	3.364 E-2	8.058 E-4	1.632 E-3	4.324 E-3	7.036 E-3
	α=1.1	1.807 E-4	5.068 E-4	4.651 E-3	2.789 E-3	7.085 E-4	1.457 E-3	3.097 E-3	5.891 E-3
PBI (θ=5)	α=1	0.963 E-4	3.352 E-4	3.931 E-3	1.267 E-2	4.368 E-4	1.249 E-3	2.362 E-3	5.326 E-3
	α=1.01	1.495 E-3	6.372 E-4	5.073 E-3	1.431 E-2	6.406 E-4	1.437 E-3	2.778 E-3	6.005 E-3
	α=1.1	1.237 E-4	2.047 E-4	4.685 E-3	1.143 E-2	5.023 E-4	1.401 E-3	2.234 E-3	5.568 E-3

It is interesting to investigate the behavior of the different scalarizing functions on DTLZ1 and DTLZ2, according to the number of objectives. According to the results provided in Table 4.30 which are in agreement with the results obtained by the real-world problem (see section 4.2), the performance of the weighted Tchebycheff scalarizing function is dependent on the value of α. In both test problems, DTLZ1 and DTLZ2, poor results were obtained when using the weighted Tchebycheff with α = 1, while good results were provided if this value is set to 1.01 and 1.1. Moreover, using the PBI function, the parameter α has a variable behavior. In DTLZ1, for example, better results were generally got with α = 1.0 and α = 1.1. However, unlike the results obtained by the real world problem, for both test functions (DTLZ1 and DTLZ2), good ones were provided by using both the PBI function with θ = 5 and the weighted sum. In fact, the PBI with θ = 0.01 is often the worst. To sum up, it is preferable to use either the weighted Tchebycheff with α = 1.1 or the PBI with θ = 5 for both DTLZ1 and DTLZ2. Obviously, the weighted sum is more suitable for DTLZ2.

4.3.4 Testing the effect of using neighborhood mating restrictions and adaptive recombination operators

In this section, we examined the effect of using neighborhood mating restrictions and adaptive operators (see section 3.3.2.1 for more details). The used scalarizing function (for MOEA/D and MOEA/DD) is PBI (5.0). The performance of each algorithm is evaluated using the average HV over 15 runs. The size of the population is set to 1000. The number of generations varies from 600 to 1000 for DTLZ1 and from 350 to 750 for DTLZ2. The mutation probability is 1/500 and the recombination probability is 0.8. The number of reference points is set to 250. The obtained results are shown in tables 4.31 and 4.32.

Table 4.31 IGD median values on DTLZ1 and DTLZ2 obtained using neighborhood mating restrictions and adaptive recombination

Test Problem	Operators	Objective Number	Number of generations	MOEA/D(PBI)	MOEA/DD	NSGA-III	Two_Arch2
DTLZ1	Bit-flip mutation / SBX crossover	5	600	3.179 E-4	2.635 E-4	5.116 E-4	6.034 E-4
				6.372 E-4	2.916 E-4	5.980 E-4	6.372 E-4
				1.635 E-3	3.109 E-4	5.979 E-3	6.967 E-4
		10	1000	3.872 E-3	1.828 E-3	2.215 E-3	2.368 E-3
				5.073 E-3	2.225 E-3	3.462 E-3	2.962 E-3
				6.132 E-3	2.467 E-3	6.869 E-3	3.101 E-3
	Using neighborhood mating restrictions and adaptive operators	5	600	3.129 E-4	1.436 E-4	6.302 E-4	1.309 E-4
				6.324 E-4	2.883 E-4	0.931 E-3	6.344 E-4
				2.654 E-3	6.317 E-4	3.364 E-3	7.038 E-4
		10	1000	1.703 E-3	1.784 E-3	2.718 E-3	1.384 E-3
				4.856 E-3	2.021 E-3	3.146 E-3	2.875 E-3
				8.032 E-3	5.033 E-3	7.037 E-3	5.031 E-3
DTLZ2	Bit-flip mutation / SBX crossover	5	350	1.219 E-3	1.128 E-3	4.254 E-3	4.203 E-4
				1.437 E-3	1.291 E-3	4.982 E-3	4.760 E-4
				1.727 E-3	1.424 E-3	5.862 E-3	4.992 E-4
		10	750	2.474 E-3	3.223 E-3	1.350 E-2	1.104 E-2
				2.778 E-3	3.752 E-3	1.528 E-2	1.459 E-2
				3.235 E-3	4.145 E-3	1.697 E-2	1.858 E-2
	Using neighborhood	5	350	2.097 E-4	1.783 E-4	0.341 E-4	1.961 E-4
				5.346 E-4	3.245 E-4	2.996 E-4	4.643 E-4

	mating restrictions and adaptive operators			4.963 E-3	3.978 E-3	4.032 E-3	7.784 E-4
		10	**750**	1.783 E-4	0.593 E-3	4.098 E-3	0.095 E-2
				2.432 E-3	3.498 E-3	1.327 E-2	1.216 E-2
				6.037 E-3	7.034 E-3	5.367 E-2	6.367 E-3

From the results presented in Table 4.31 which are in accordance with those provided by the real world problem, we notice that the use of neighborhood mating and adaptive operators with different number of objectives improve considerably the search performance. For DTLZ1 (DTLZ2, respectively), best IGD median values were obtained using MOEA/DD (MOEA/D, respectively) especially with a high number of objectives. However, MOEA/D improved considerably the average IGD values, compared to other algorithms. Moreover, our results prove that the diversity was enhanced without deteriorating the convergence when mating similar parents.

Table 4.32 IGD median values on DTLZ(2,15) and DTLZ(3.15) provided by applying neighborhood mating restrictions and adaptive recombination operators

Test Problem	Operators	MOEA/D(PBI)	MOEA/DD	NSGA-III	Two_Arch2
DTLZ5(2,15)	Bit-flip mutation / SBX crossover	1.134 E-2	1.641 E-3	2.034 E-3	1.344 E-3
		1.326 E-2	3.195 E-3	4.985 E-3	2.786 E-3
		1.667 E-2	4.034 E-3	9.031 E-3	4.239 E-3
	Using neighborhood mating restrictions and adaptive operators	1.068 E-2	1.327 E-3	2.031 E-3	2.323 E-3
		1.261 E-2	2.857 E-3	3.934 E-3	2.407 E-3
		1.661 E-2	4.906 E-3	9.017 E-3	7.301 E-3
DTLZ5(3,5)	Bit-flip mutation / SBX crossover	1.194 E-4	0.634 E-4	1.985 E-4	0.736E-4
		1.265 E-4	3.056 E-4	3.192 E-4	0.925E-4
		1.703 E-4	6.238 E-4	8.054 E-4	2.362E-4
	Using neighborhood mating restrictions and adaptive operators	0.806 E-4	1.038 E-4	1.964 E-4	0.852 E-4
		1.103 E-4	3.042 E-4	3.036 E-4	0.863 E-4
		1.698 E-4	6.179 E-4	8.031 E-4	5.296 E-4

According to the table 4.32, the improvement on the IGD values after including the reduction is clearer in DTLZ(2.15) than DTLZ(3.5). Surprisingly, in DTLZ5(3,5), the MOEA/D is more efficient than other algorithms when using neighborhood mating restrictions, adaptive operators and correlated objectives .

4.3.5 Testing the effect of hybridizing the EMOs with the dimensionality reduction approach

In this section, we test the effect of adding the proposed dimensionality reduction approach. The used scalarizing function (for MOEA/D and MOEA/DD) is PBI (5.0). The mutation probability is 1/400 with an index of 50 and the recombination probability is 0.9 with an index of 5. Neighbor mating and adaptive recombination is used. The size of the population is set to 1000. We employ 250 reference points for NSGA-III.

Overall, the detailed results of applying the proposed dimensionality reduction approach (The NL-MVU-PCA algorithm for non-linear objective reduction and L-PCA for linear objective reduction) to the DTLZ problems are not presented in this section. More details of its application on DTLZ and WFG problems can be found on (Saxena et al. 2013). Besides, the NL-MVU-PCA and L-PCA does not suggest a reduced set of objectives because the DTLZ2 is known by a PF that involves all the objectives (Sinha et al. 2013). Thus, our work only presents (Table 4.33), the smallest non-redundant objective set and the needed number of iterations of the different problems such as DTLZ5 (2, 15) and DTLZ5 (3, 5) to evaluate the proposed algorithm.

Table 4.33 The smallest non-redundant objectives and the needed iteration number for the proposed algorithms on DTLZ5(2,15) and DTLZ5(3,5)

	DTLZ5(2,15)		DTLZ5(3,5)	
	Smallest non-redundant objectives (average)	Needed iteration number (average)	Smallest non-redundant objectives (average)	Needed iteration number (average)
MOEA/D(PBI)	2	3684	3	1875
MOEA/DD	2	3256	3	1822
NSGA-III	2	3894	3	1923
Two_Arch2	2	3866	3	1934

Although the obtained results in Table 4.33 shows that all tested algorithms are able to identify the smallest non-redundant objective set, the algorithms differ in the number of the needed iterations used to find the smallest set. For example, in the DTLZ5(2,15) problem, Two_Arch2 needs to perform 3866 iterations to find the smallest non-redundant objective set, while MOEA/DD requires only 3256 iterations.
Moreover, in order to investigate the quality of the obtained solution after adding the proposed dimensionality reduction approach, we present the median IGD values before and after applying the dimensionality reduction on the proposed algorithms.

Table 4.34 median IGD values before and after applying the dimensionality reduction

Test Pb	Dimensionality Reduction	MOEA/D(PBI)	MOEA/DD	NSGA-III	Two_Arch2
DTLZ5(2,15)	Without reduction	1.134 E-2	1.641 E-3	2.034 E-3	1.344 E-3
		1.326 E-2	3.195 E-3	4.085 E-3	2.786 E-3
		1.667 E-2	4.034 E-3	9.031 E-3	4.239 E-3
	Using L-PCA / NL-MVU-PCA	0.962 E-2	2.498 E-3	3.262 E-3	2.093 E-3
		0.967 E-2	2.632 E-3	3.647 E-3	2.243 E-3
		1.460 E-2	6.703 E-3	4.053 E-3	2.511 E-3
DTLZ5(3,5)	Without reduction	1.194 E-4	0.634 E-4	1.985 E-4	0.736E-4
		1.265 E-4	3.056 E-4	3.192 E-4	0.925E-4
		1.703 E-4	6.238 E-4	8.054 E-4	2.362E-4
	Using L-PCA / NL-MVU-PCA	0.934 E-4	2.838 E-4	2.360 E-4	0.602 E-4
		1.027 E-4	2.923 E-4	2.786 E-4	0.659 E-4
		1.175 E-4	3.037 E-4	2.837 E-4	1.231 E-4

Table 4.34 demonstrates that the IGD values decrease when the dimensionality reduction is used, which reveals the effectiveness of using the dimensionality reduction. Regarding the improvement rate before and after the reduction, the MOEA/DD surpasses other algorithms for DTLZ5(2,15), while the NSGA-III surpasses other algorithms for DTLZ5(3,5).

4.3.6 Testing the effect of hybriding the EMOs with dimensionality reduction and user preferences

In this section, we investigate the effect of incorporating both dimensionality reduction and user preferences in our approach. The used scalarizing function (for MOEA/D and MOEA/DD) is PBI (5.0). The mutation probability is 1/400 with an index of 50, and the recombination probability is 0.9 with an index of 5. The size of the population is set to 1000. Neighbor mating and adaptive recombination are used. We apply 250 reference points for NSGA-III. Preference approaches suppose that no redundant objective exists in the given problem (Saxena et al. 2013). Thus, dimensionality reduction procedure is always performed before applying the DM preferences.
After employing the reduction approach, the preference is applied on a reduced set of objectives. Table 4.35 shows the final solutions characteristics on the DTLZ5(3,5) problem. Each run has a different initial population which is the result of applying our reduction procedure on the concerned EMO. As mentioned before, $\mathbf{d_s}$ (the expected improvement in solutions from the current best solution) = 0.01 and **TDMax** (the maximum number of calls of the preference information from the DM) = 30.

Table 4.35 Median distance between the obtained solutions and the most preferred solutions for d_s =0.01

	MOEA/D	NSGA-III	Two_Arch2	MOEA/DD
Accuracy	0.203	0.396	0.409	0.009
Number of function evaluations	6132	8038	8036	5981
Number of required DM calls	19	26	25	22

4.4 Simulations: Modeling the used protocols, simulation with small and large instances

4.4.1 Network protocol modeling

4.4.1.1 Modeling the 3D location

Although 3D location is more realistic and accurate, most existing location systems focus on the 2D plane. Like 2D localization, two types of location can be distinguished in 3D environments: range-based localization and range-based localization. This classification is based on the fact that the actual distance between the nodes must be measured or not. Free-range 3D locating algorithms measure the exact orientation and distance between neighboring nodes, and use this information to locate the nodes. While locating algorithms based on free range have a certain hardware requirement. Therefore, they are more expensive to implement in practice. The Landscape 3D algorithm (Zhang et al. 2006) for example, is a 3D location algorithm according to the scope. Different 3D localization algorithms exist as free range 3DDVHOP algorithm (He et al. 2003), the 3D centroid algorithm (Bulusu et al. 2000) and 3D MDS-MAP (Tan et al. 2010).

To compensate for the deficit of the two types of localization algorithms mentioned, we propose to implement a localization based on rssi and iron exchanges hybridized with the DV-HOP protocol. Indeed, we propose a hybrid localization method based on the combination of the jump distance and the RSSI data for the 3D location. We propose a hybrid localization system that improves the free-range location technique (DV-HOP) in 3D by introducing an RSSI datum to revise the jump distance. First, 3D DV-HOP uses network connectivity information to estimate node locations in 3D space. Then, the RSSI value can be easily collected to correct the position of the nodes found by the DV-HOP depending on the signal strength. The RSSI value is measured in both directions: from the mobile node, we measure the received RSSI value of the other nodes (fixed, nomad or mobile); And from each fixed, nomad or mobile node, the RSSI value received from the mobile node is measured. The final permissible value of the RSSI between the mobile node and each other node corresponds to the higher value of the two values already mentioned.

Indeed, the different localization algorithms require that each normal node (which is not anchored) must have at least three neighboring anchor points, the 3DDV-Hop algorithm does not require 3 neighbor anchors per node. When applying the DVHOP algorithm, we need three nodes (which are not necessarily anchors) to locate each node. In our proposed model (3DDVHOP + RSSI), four nodes are required: three nodes of the 2D plane that the DVHOP algorithm requires and a node for the height. Figure 4.1 shows the location using the 3DDVHop and RSSI algorithm with a network topology with 3 anchors and 4 normal nodes. A1, A2 and A3 are anchors whereas Nx, N1, N2, N3 are normal nodes.

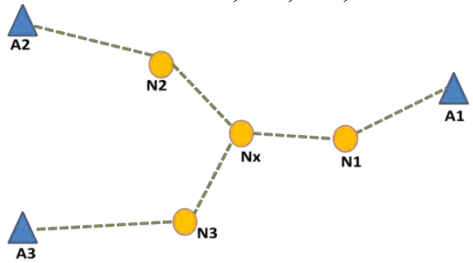


Figure 4.1: localization algorithm with 3D DV-HOP + RSSI

The RSSI value can be influenced by various factors other than distance, such as node movement, environment, antenna orientation, or energy level. Thus, a detailed modeling of the physical environment, data binding and routing, greatly improves the obtaining of a precise distance from the RSSI value.

4.4.1.2 Physical layer (radio)

In our simulations, the physical layer uses a frequency set at 433 Mhz. Other parameters are initialized with significant values. Some values are set according to standard protocols such as 802.15.4 and some other values are derived from empirical values of our experiments**.**
The following parameters are considered in our simulations:

- Transmit power = default (60 mW);
- Reception gain = default (50 mA);
- Operating temperature = default (25 c);
- Bit rate = default (256 kbps);
- RSSI = by default (140);
- Bit error rate = default (0.01);
- Range inside = by default (15m);
- Maximum number of nodes = by default (65000) according to protocol 802.15.4;
- Power supply = default (3.6 v);

4.4.1.3 Data Link Layer: CSMA / CA of 802.15.4

IEEE 802.15.4 is considered a promising technology for WSN, and a great deal of research has been conducted to study the access and detection of collision avoidance multiple access (CSMA / CA) carrier in 802.15 protocol .4. When more than one station tries to transmit one frame at a time, a collision occurs, and eventually all data can be damaged. The standard mechanism for resolving conflicts in computer networks is called multiple access carrier (CSMA) detection. Indeed, the CSMA algorithms attempt to break the symmetries of faulty transmissions restarting at the same time, using random binary explicit backoff procedures. While wired devices can listen during their own transmissions and use CSMA with collision detection (CSMA / CD), stations in wireless networks generally cannot listen to their own broadcasts, and therefore collision transmissions cannot be detected after they have been completed. Thus wireless devices use CSMA with collision avoidance (CSMA/CA or CSMA-CA). For the data link layer, a problem of access priority to the channel arises if two nodes within range (of them) speak simultaneously. We are interested in the modeling of the 802.15.4 CSMA / CA protocol. A non-coordinated CSMA is used (not slotted, without acknowledgment and without RTS/CTS).

4.4.1.4 Routing Layer: Reactive AODV

To model the routing layer, the reactive AODV protocol is used. The AODV (Ad-hoc on request Distance Vector) is a routing protocol for ad-hoc networks. It is designed to self-start in a mobile node environment, and to withstand a variety of network behaviors such as node mobility, link failure, and packet loss. At each node, the AODV maintains a routing table. The entry of the routing table for a destination contains three essential fields: the next hop node, a sequence id and a hop count. All packets destined for the destination are sent to the next hop node. The sequence number acts as a timestamp form, and is a measure of the freshness of a route. The number of hops represents the current distance to the destination node. In the AODV, nodes discover roads in demand-response cycles. A node requests a route to a destination by broadcasting a RREQ message to all its neighbors. When a node receives a RREQ message but does not have a route to the requested destination, it in turn transmits the RREQ message. This node also remembers a path inversion to the requesting node, which can be used to transmit subsequent responses to this RREQ. This process repeats until the RREQ reaches a node that has a valid route to the destination. This node (which can be the destination itself) responds with a RREP message. This RREP is a unicast along the reverse routes of the intermediate nodes until it reaches the originating node. Thus, at the end of this request-response cycle, a bidirectional path is established between the requesting node and the destination. When a node loses connectivity to its next hop, this node cancels its route by sending a RERR to all nodes that have potentially received its RREP.

In our case, each node, by receiving a RREQ, it returns it to all these neighbors even the source that absorbs. A TTL is also used which is the maximum time at the end of which a new RREQ with a higher sequence number is sent if no response is received from the destination node or a node that knows a valid path to the destination. The destination receiving the RREQ chooses a new source from these neighbors; this destination node becomes the source and starts a new cycle by sending a new RREQ message. This process of request-response cycles stops after 4280 seconds of simulations (~72 minutes).

4.4.1.5 Energy modeling and network lifetime

A deployed active sensor dissipates energy during transmission, detection, reception of a message or period of inactivity. Therefore, energy efficiency is an essential key in designing a network of wireless sensors. The optimization of energy consumption then becomes an essential factor for the optimization of the operation of a WSN. To model the energy consumption in our simulations, using an initial energy E o indicator in each node. Given that following the 802.15.4 protocol, the reception is more expensive than sending energy, E_0 will be reduced by 'i' units if you send a message, and it will decrease $2\times i$ units if it receives a message. The energy consumption is linked to another indicator which is the lifetime of the network. Typically, the lifetime of the WSN is represented by the time after which the first node is out of service. Thus, for each variant of the algorithm, the time corresponding to the lifetime of the network is measured so as to have an idea of the influence of the choice of the locations of the new nomad nodes added over the lifetime of the network.

4.4.2 Small-scale simulations of the network with OMNet ++

In our simulations, we use OMNeT ++ 4.6 (Omnetpp 2018) which is a free platform for simulation and development of network protocols. OMNeT ++ IDE is based on the Eclipse platform and contains a simulation kernel library, a topology description language called NED, and a graphical user interface for running the simulation, it also provides a component architecture For modules programmed in C ++.

For small-scale simulations, the 3D case is considered. Four objectives were considered incrementally (two objectives, then three objectives, then four objectives): cost in nodes, coverage, location, and optimization of energy consumption.

In small scale simulations, we try to imitate our experiments conducted at the IUT University (see section 4.5)**.** The network architecture is based on a fixed number (11) of nodes having known positions. The number of nomad nodes to be added is also fixed at three. Their positions are determined with the algorithms tested. Similarly for mobile nodes, only one node is used as trigger for the first message. The positions of the initially deployed fixed nodes are chosen according to the distribution law used by OMNet ++ which tries to distribute uniformly the nodes from the center of the region of interest, this leads to the non-coverage of certain zones at the borders if the number of nodes Fixed is too small.

The runtime scenario is: An initial message is sent from the mobile node to a random destination 'd'*;* once d is found by the AODV routing protocol, *'d'* becomes the source and a new destination is selected ... etc. This cycle is repeated until a stop condition is satisfied, among other things a maximum simulation time. In order to imitate our experiments, the small scale simulations follow a fixed scenario in which the number of nodes is fixed;

For connectivity to nodes in this small scale scenario, a connectivity matrix is used between nodes that are deducted from the empirical results derived from our experiments (Mnasri et al, 2017a). The same connectivity links of the experiments are thus used initially. Subsequently, in order to model the dynamism of the network, these connectivity links are set to a perturbation which makes it possible to modify these links of initial connectivity. This perturbation concerns the calculation of the RSSI rates between the nodes. Indeed, a matrix of RSSI rates extracted from experiments is used initially. Then, this matrix is set to a

perturbation (+/- 30 for each value) in order to have new connectivity relations between the nodes.

In the case of small scale simulations, we adopt the same architecture (number and type of nodes) 3D deployment used in the experiments (see Figures in the section 4.5).

Figure 4.2 represents the distribution of the nodes according to the OMNet++ interface in the small-scale simulations. Nodes named initial-i represents the fixed nodes and nodes named nomad-i represents the nomad nodes.

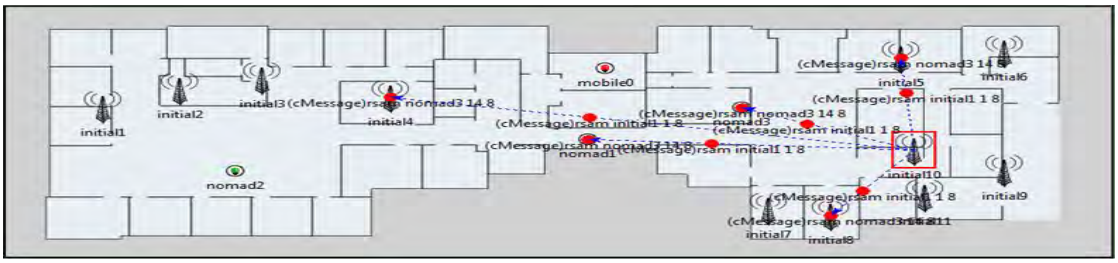


Figure 4.2 small-scale simulation scenario

4.4.2.1 Variations in RSSI rates in small-scale simulations

To measure the location, we use the RSSI metric since the localization model used is based on a hybridization between the RSSI and the DVHop protocol. As a result, the higher the RSSI, the better the localization. A neighbor can enter the neighborhood table of a node only if the RSSI value of the detected node is greater than a predefined threshold. The theoretical value of this threshold is set at 100. Initially, RSSI levels are based on our empirical experiments and to guarantee dynamism within the network, disruption of the value of RSSI is introduced via A random function (+ -30).

Figure 4.3 shows, for different number of objectives considered by the algorithms tested, the average of the RSSI rates measured for all the nodes in connection with (detected by / detecting) the mobile node.

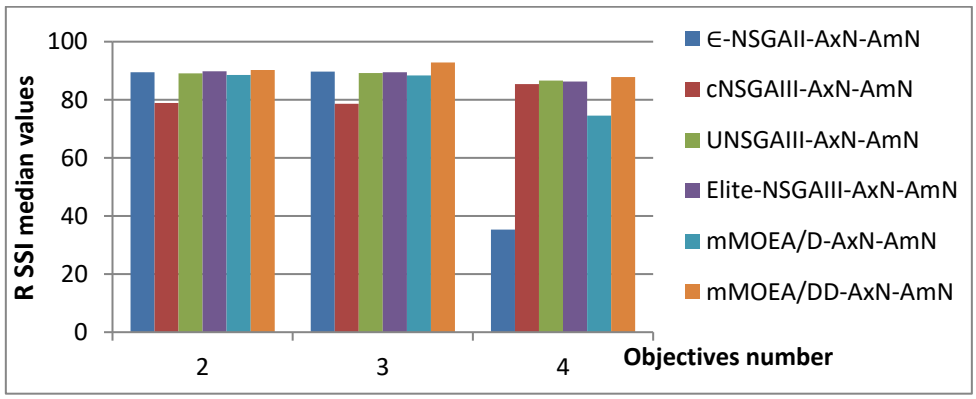


Figure 4.3 Average RSSI rates during small-scale simulations

4.4.2.2 Variations in FER rates in small-scale simulations

To measure coverage, we use the Frame Error Rate (FER) as a measure to assess the quality of links between nodes. As a result, the lower the RMF, the better the coverage. Although the values of the FER are less variable than those of the RSSI, we take, for each pair of nodes (node i - node C); i ∈ [1, 14]; an average value extracted from four values taken with an interval of 10 seconds between the two. Initially, FER rates are based on our empirical experiments and to guarantee a dynamism within the network, disruption of the value of the FER is introduced via a random function (+ -0.04 to + - 0.2). Figure 4.4 shows the mean FER rates measured for all the nodes in connection with (detected by / detecting) the mobile node.

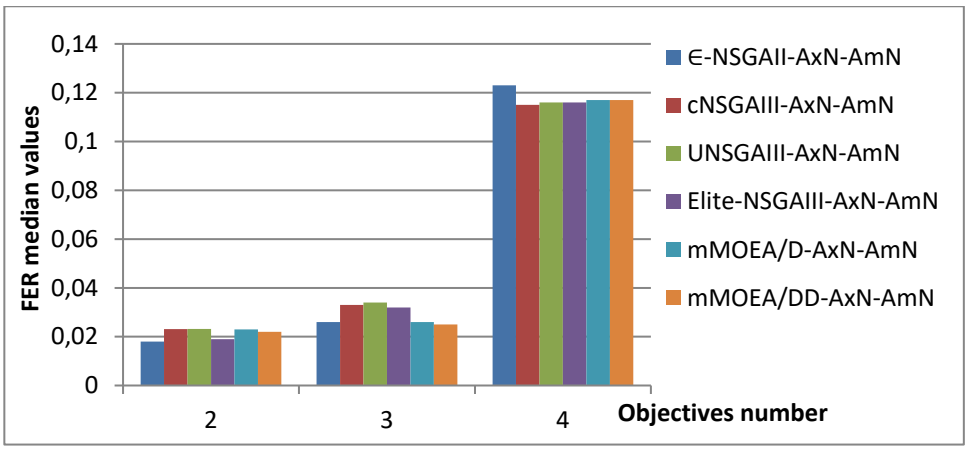


Figure 4.4 Average FER rates in small-scale simulations

4.4.2.3 Variations in the number of neighbors in small-scale simulations

To get an idea about the connectivity of the network, one measures the number of neighbors of the target (the mobile node). Figure 4.5 represents the average of the number of neighbors of the mobile node per objective.

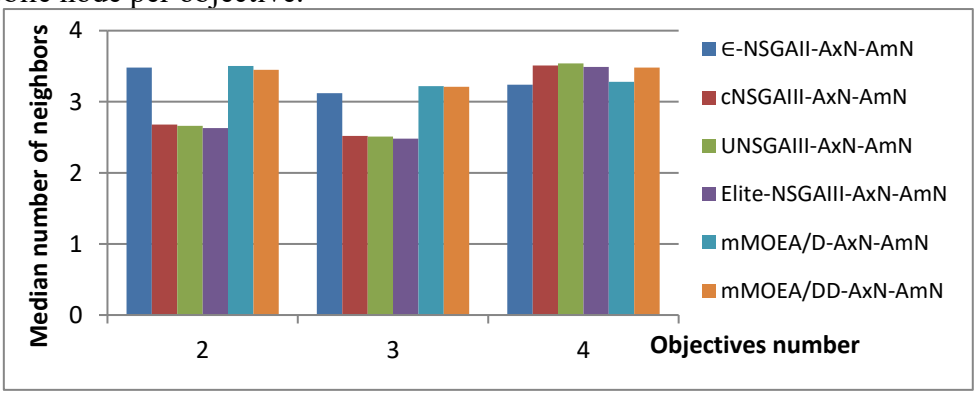


Figure 4.5 Average number of neighbors in small-scale simulations

4.4.2.4 Variations in energy and network lifetime in small-scale simulations

Figures 4.6 and 4.7 show the variations of the energy level of the network in the small-scale simulations for different number of objectives. Indeed, for the different algorithms tested, an average of the energy indicator of the nodes of the network is measured after the addition of the nomad nodes, this according to the number of fixed nodes.

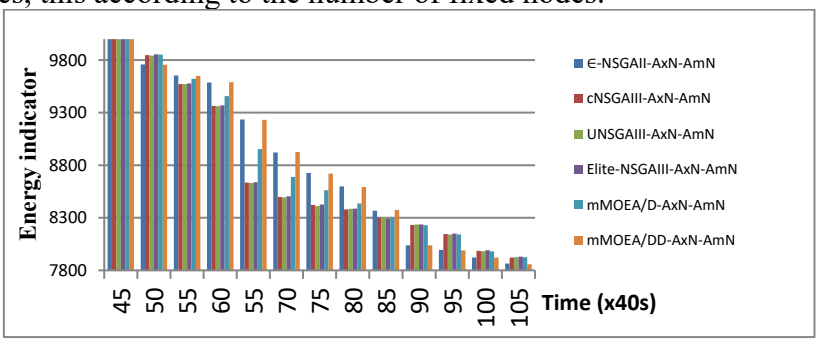


Figure 4.6 Variation of energy in relation to the number of fixed nodes, for 2 objectives

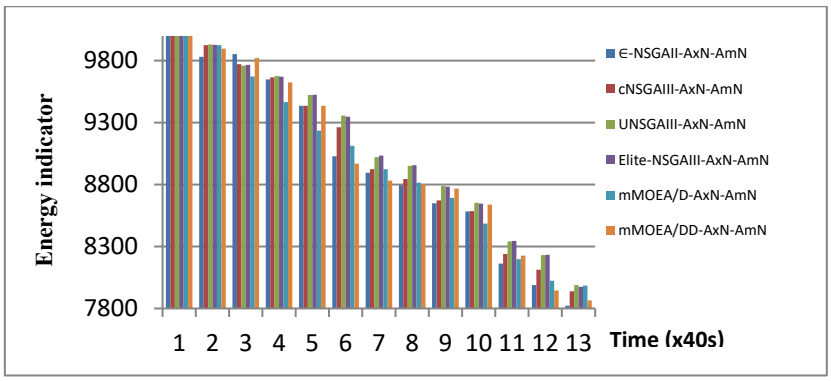


Figure 4.7 Variation of energy in relation to the number of fixed nodes, for 4 objectives

Figure 4.8 shows the time after which the first node of the network is switched off for different number of objectives, which gives us an idea of the lifetime of the network.

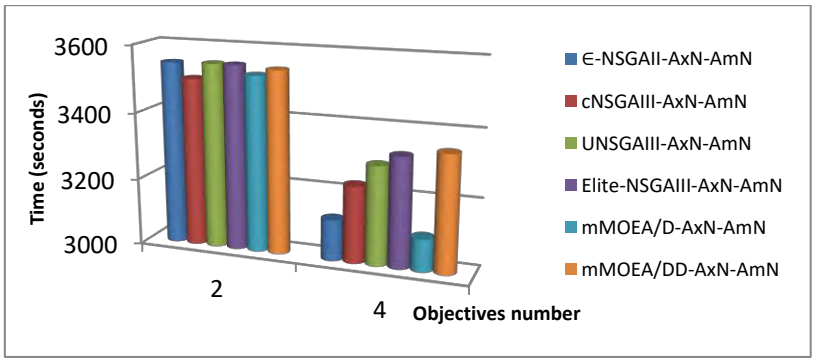


Figure 4.8 Change in lifetime in relation to the time

4.4.2.5 Comparison of experiments and simulations for small scale instances

Figure 4.9 illustrates similar results of experiments and simulations for small scale instances.

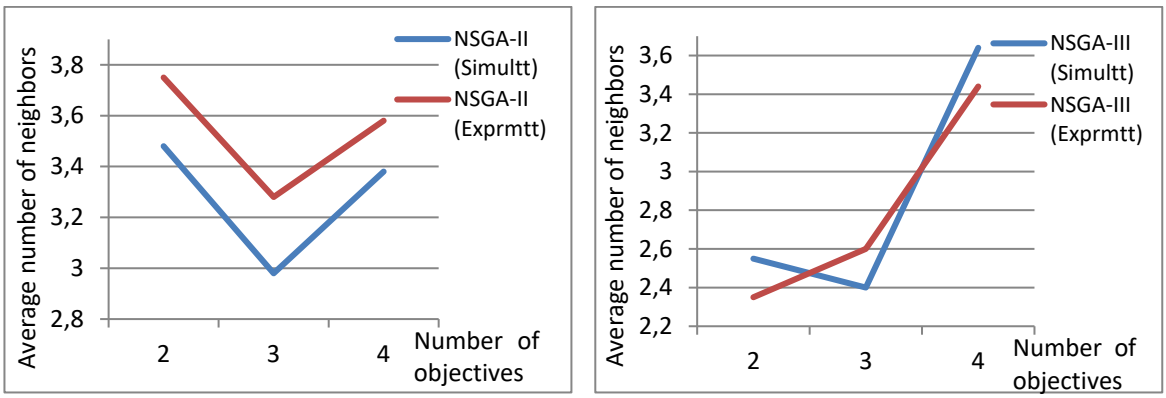


Figure 4.9 Experiments vs. simulations for small scale instances

4.4.3 Large scale network simulations with OMNet++

Large-scale simulations are carried out in order to prove the scalability and robustness of the proposed approach. Like simulations on a small scale, the OMNet++ 4.6 simulator is used. The 3D case is considered. The same objectives and the same execution scenario are used. In contrast, the network architecture, connectivity, range and mining with which the RSSI are different from those used in small-scale simulations.

The network architecture used is based on a number of fixed nodes determined by the user at the beginning of the simulations. The positions of these fixed nodes are random and chosen by the OMNet++ simulator. In our simulations, we usually choose more than 100 fixed nodes. For the nomad nodes to be added, their number is variable and fixed by the

algorithms tested. This number depends on the region of interest, the number of nodes initially deployed and the optimization algorithm tested. For mobile nodes, a single node is used to trigger the simulations.

Connectivity in large-scale simulations is more dynamic than that of small-scale simulations. It is deduced from the radio range of the nodes. This range can be either fixed for all nodes, or deduced from the transmit power and gain. In our simulations, we use the second alternative based on transmission power and gain. To calculate RSSI, using the formula of frees which include the distance and power:

$$RSSI(d)\left[dBm\right] = RSSI_{ref} - 10n\log_{10}(d/d_{ref})$$

Figure 4.10 shows an example of large-scale simulation with 85 fixed nodes, 36 nomad nodes, and a mobile trigger node.

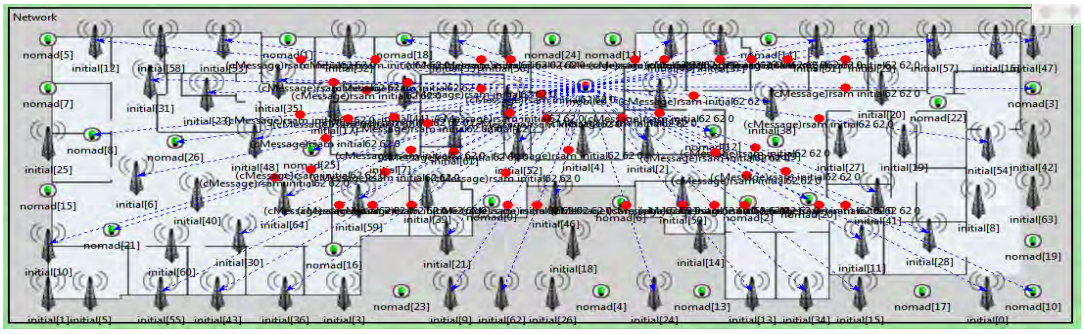


Figure 4.10 Large-scale simulation scenario (85 fixed nodes and 36 nomad nodes)

4.4.3.1 Variations in the number of nomad nodes added in large-scale simulations

Figures 4.11 and 4.12 show the variations in the average number of nomad nodes added to the network in the small-scale simulations for different number of objectives. Indeed, for the different algorithms, an average of the necessary number of nomad nodes to be added to the network is measured, according to the number of fixed nodes.

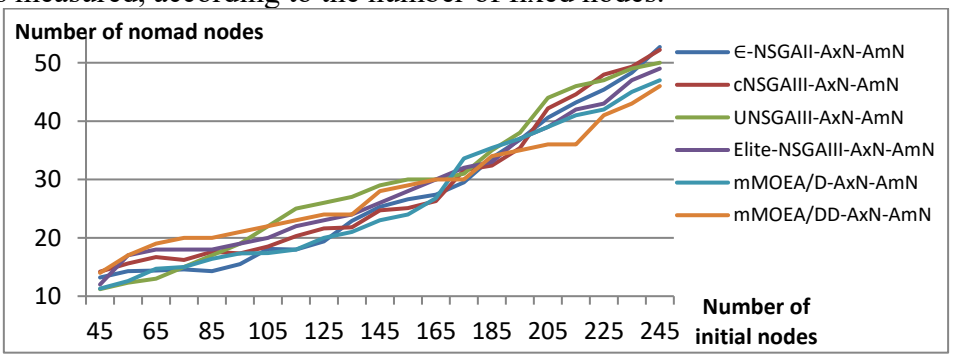


Figure 4.11 Variations in the number of nomad nodes added for two objectives

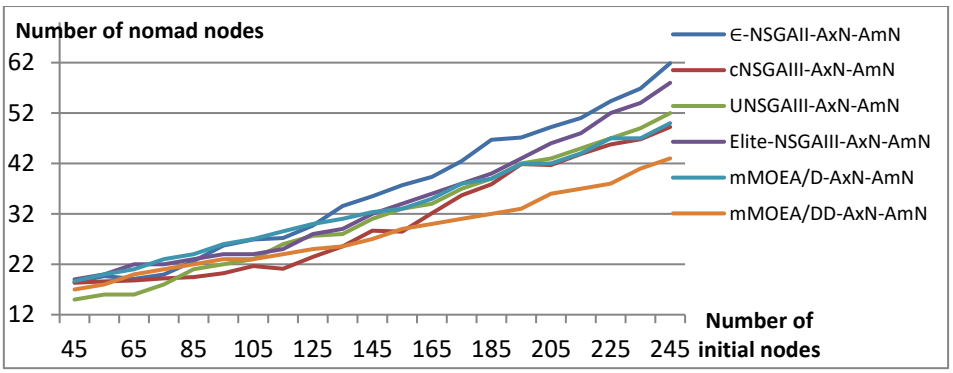


Figure 4.12 Variations in the number of nomad nodes added for four objectives

4.4.3.2 Changes in RSSI rates in large-scale simulations

Figures 4.13 and 4.14 represent, for different number of objectives considered by the different algorithms, the average of the RSSI rates measured for all the nodes in connection with (detected by / detecting) the mobile node.

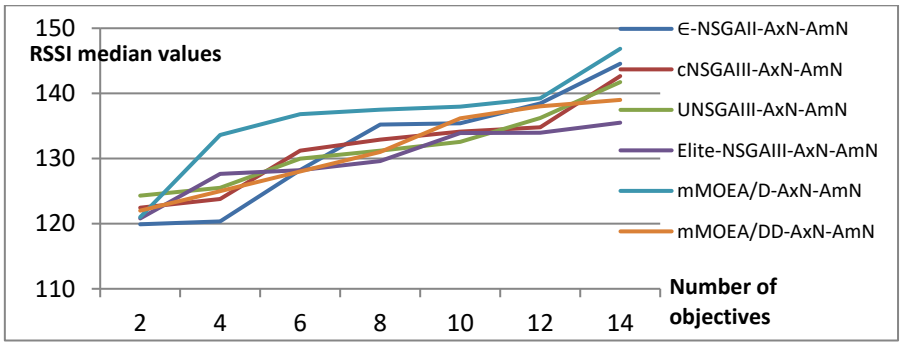


Figure 4.13 Average RSSI rates for different number of objectives

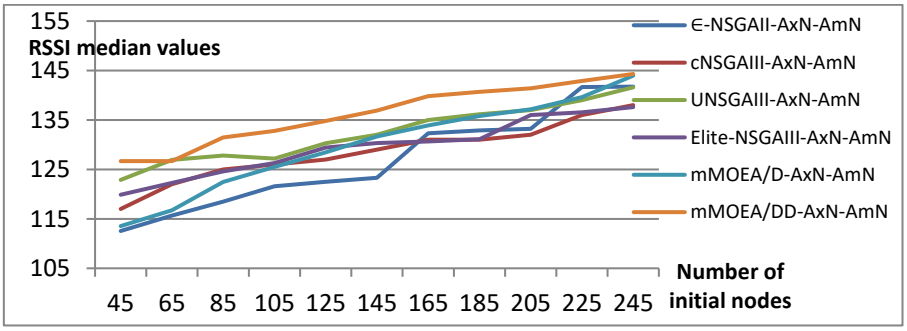


Figure 4.14 Average RSSI rates according to the number of initial nodes

4.4.3.3 Variations in FER rates in large-scale simulations

Figures 4.15 and 4.16 show, for different number of objectives considered by the various algorithms, the FER average rates measured for all the nodes in connection with (detected by/detecting) the mobile node.

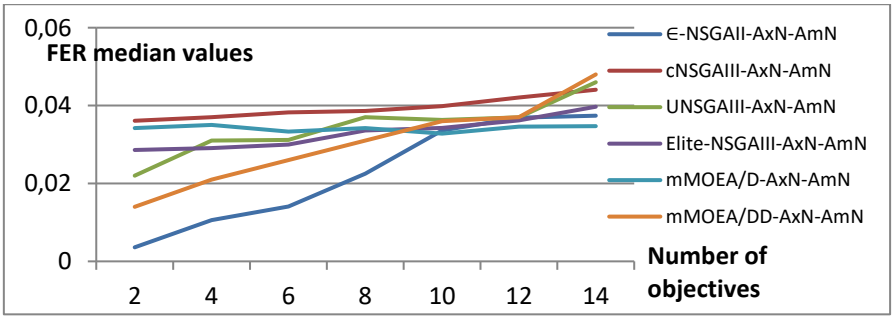


Figure 4.15 Average FER rates for different number of objectives

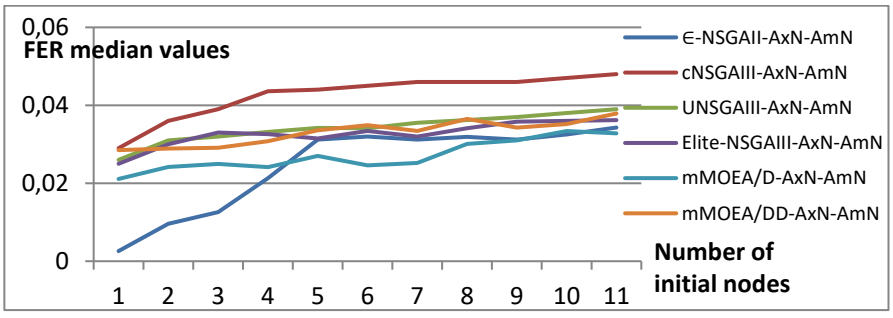


Figure 4.16 Average FER rates according to the number of initial nodes

4.4.3.4 Variations in the number of neighbors in large-scale simulations

Figures 4.17 and 4.18 represent, for different number of objectives considered by the various algorithms, the average numbers of neighbors of the mobile node.

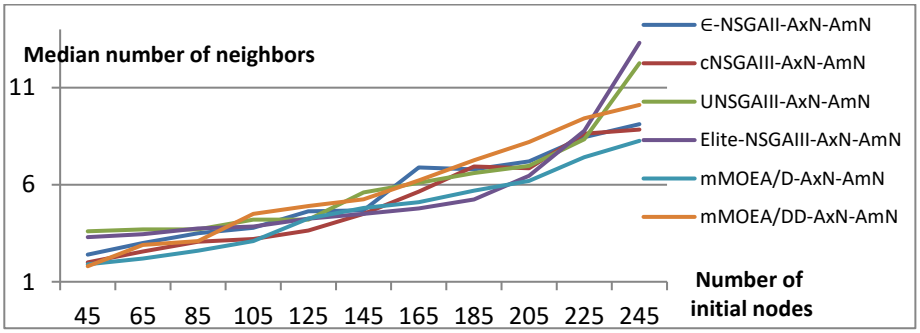


Figure 4.17 Average numbers of neighbors for three objectives

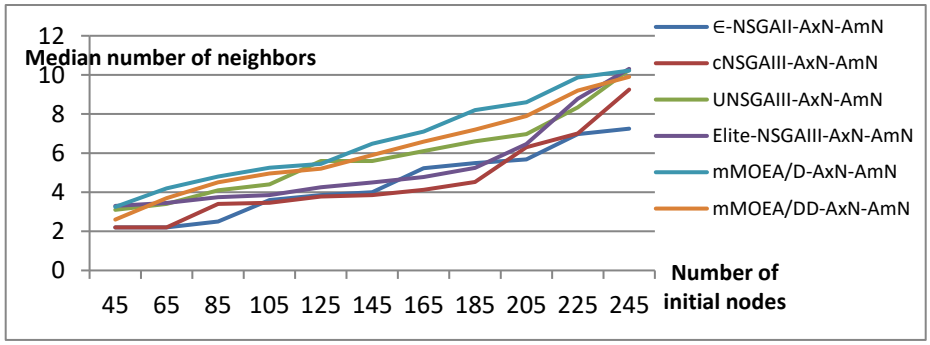


Figure 4.18 Average numbers of neighbors for nine objectives

4.4.3.5 Variations in energy consumption and network lifetime in large-scale simulations

Figures 4.19 and 4.20 represent the variations of the energy level of the network in the small-scale simulations for different number of objectives. Indeed, for the various algorithms, one measures an average of the energy indicator of the nodes of the network after the addition of the nomad nodes, according to the number of fixed nodes.

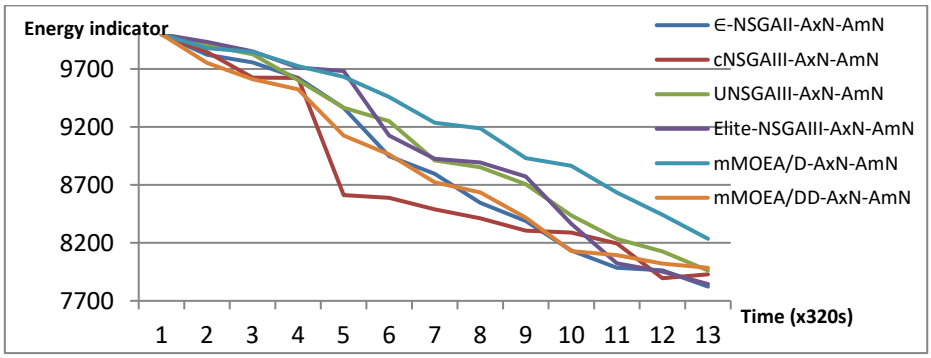


Figure 4.19 Variation of energy in relation to the number of fixed nodes, for two objectives

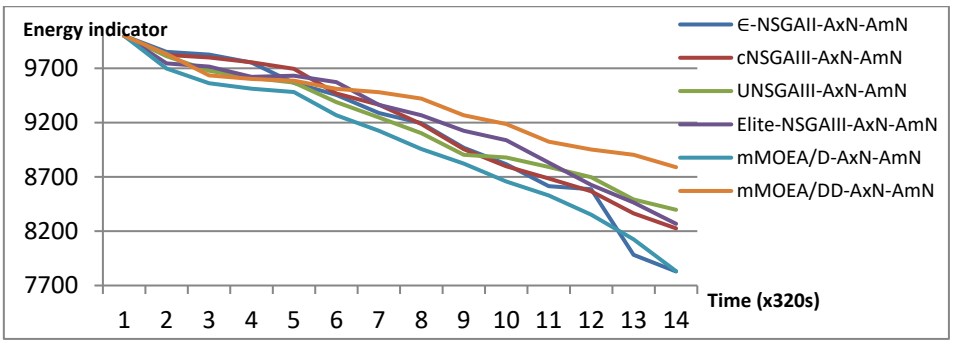


Figure 4.20 Variation of energy in relation to the number of fixed nodes, for four objectives

Figure 4.21 shows the time after which the first node of the network is switched off for different number of objectives, which gives us an idea of the lifetime of the network.

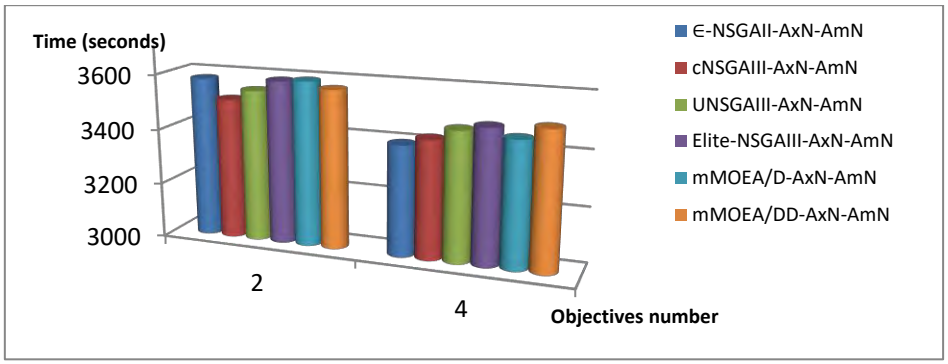


Figure 4.21 Change in lifetime in relation to the number of targets

4.4.4 Discussion and interpretations

After the analysis of the simulations, different interpretations can be envisaged:

- In some cases, lower RSSI averages are recorded after adding the nomad nodes. Despite this decrease indicating that the RSSI rates of the new (nomad) nodes are lower than the RSSIs of the installed (fixed) nodes, the location rate, the coverage rate and the number of neighbors are improved. Given the objectives set by our approach, this decrease in RSSI averages is understandable, since adding a node in a x_1 location so that it will be close to several nodes with a lower RSSI value will be better than adding it in a location x_2 with a higher RSSI value but a smaller number of neighbors.
- By comparing small and large scale simulation results, large-scale simulation results are more stable (at the level of convergence and the monotony of the curves) and give a clearer idea of the efficiency of the simulations. New deployment positions proposed in the improvement of objectives. This is explained by the fact that large-scale simulations give the possibility of having a larger population of individuals, which improves the selection of the best individuals thereafter. Similarly, mutation and crossing operators have the option of covering a larger search space, which allows identifying the most ideal solutions (deployment locations).
- Simulation results in small scale (Figures 4.3, 4.4, 4.5 and 4.6) show conformity with the experimental results during the day, the level of increased coverage and localization, which proves the effectiveness of the proposed approach in different contexts.
- By comparing the efficiency of the different algorithms tested, the simulation results show that this efficiency is relative to the number of objectives to be met. For example, Figures 4.11 and 4.12 (also 4.15 and 4.16) show that if the number of goals is less than three, the ∈-NSGA-II is more efficient, while the NSGA-III is more effective than if ∈-NSGA-II Number of objectives exceeds three. This is explained by thefact that the ∈–NSGA-II is dedicated to multi-objective problems, while the authors of the NSGA-III propose it as an adaptation of

the NSGA-II for problems hawing more than four objectives (many-objective problems). Our study constitutes a proof by experimentation and simulations of this observation which has been proved by the authors of the tested algorithms (MOEA/DD and NSGA-III) only by tests on instances of theoretical problems (benchamarks).

4.5 Experimentations on real testbeds

To evaluate the robustness and the efficiency of the protocols, technologies and models in real environments, real platforms called testbeds can be used. Indeed, simulations and theoretical calculations fail to reproduce the physical characteristics of the real-world environments; hence the current tendency to reduce the differences between theory and practice by testing the new algorithms and solutions in real environments with experiments carried out on testbeds.

In this study, we propose a testbed composed of TeensyWiNo nodes. Using this personal testbed, different advantages are envisaged:

- Conformity to reality: A personal testbed like ours is based on tests in a real context of use unlike test platforms such as FIT/IoT-Lab (IoTLab 2017) which offers tests with a large number of nodes that are aligned or uniform on a grid.

- Reproducibility: since our testbed relies on open-source tools, such as OpenWiNo and Arduino, it is easy to manage and reproduce the obtained results by other research teams. Indeed, Ophelia supports different physical layers and various types of sensors, which facilitates the deployment of nodes and the prototyping task.

- Heterogeneity of nodes: Our testbed supports three different types of nodes (DecaWino, WiNoLoRa and TeensyWiNo). Thanks to its compliance with open hardware and software, the WiNo architecture allows integrating foreign libraries to manage the deployed nodes, which enables it to support a wide variety of nodes.

- A distributed deployment: Our testbed consists of more than 30 nodes deployed in several buildings and locations in a campus of 200 × 200 m².

- Easy use and deployment: The nodes in our testbed are manipulated (erasing data, updating) using OpenWino and the execution of the protocols stack is done via the usb interface of the nodes or by executing a command line from the console. Moreover, WiNo nodes are compatible with revolutionary transmission modes (UWB, LoRa,..etc) and most standard physical layers, which makes the design and customization of the network as well as the replacement of the physical layer easier and more realistic.

- Real life usage: WiNos nodes have small size, low power consumption rate and easy attachment to a mobile system or a person, which makes them an ideal component for the IoT and the prototyping of communicating objects.

4.5.1 Experimental parameters and used tools

An Intel Core i5-6600K 3.5 GHz computer is used to test the algorithms. Physical layer is based on a 433 MHz implementation. The applied access method is the non-coordinated CSMA/CA of the IEEE 802.15.4 protocol, and the routing layer relies on the reactive Ad-hoc On-demand Distance Vector protocol. The parameters considered in our experiments are illustrated in Table 4.36.

Table 4.36 Parameters of the experiments

Nodes repartition	6 sites on 200 * 200 m²
Nodes number	36 (29 fixed, 6 nomad, 1 mobile)
Sensing range	8m
Transmission range	7m
FER (Frame Error Rate)	0.01 (initially)
RSSI (Received Signal Strength Indicator)	100 (initially)
Average number of runs	25 experiments
Bit rate	256 kbps
Modulation model	125 kbit/s GFSK

Antenna model	transceiver RFM22
Modem configuration	12 # GFSK_Rb2Fd5
Frequency	434.79 MHz
Tx power	7 (the max of RFM22)
Message-number	1000
Message-length	16
Message-wait	5
Reception gain	50 mA
Transmission power	100 mW

The deployed TeensyWiNo nodes: are WiNoRF22 nodes equipped with brightness and temperature sensors to which other sensors are added (gyrometer, acceleration or pressure). They give access to low layers in order to manage the access time to the medium, the sleep, the awakening time, the CPU time and the management of the restricted memory. WiNo nodes represent a hardware platform able to host different protocols with real-time constraints (several months of use using two AAA batteries). The installed TeensyWiNo nodes are shown in Figure 4.22 and their technical characteristics are illustrated in Table 4.37.

Figure 4.22 The Teensy WiNo used nodes

Table 4.37 Technical characteristics of the used TeensyWiNo nodes

CPU/RAM/Flash	CPU/RAM/ Flash ARM Cortex M4 (32bit) 72MHz, 64kB RAM, 256kB Flash (PJRC Teensy 3.1)
Transceiver (Arduino libraries)	HopeRF RFM22b : 200-900MHz, 1-125kbps, GFSK/FSK/OOK, +20dBm RadioHead
Use	IoT, WSN

These nodes, incorporated in the Arduino ecosystem, facilitate the integration of hardware and software components (interaction devices, actuators, sensors, processing algorithms, etc.), which allows obtaining the feedback from the user's experience.
The following tools are used:

- **Arduino 1.6.1** (Arduino 2018): is an open software/hardware platform employed by the WiNo nodes to prototype modules and transfer the sketches via serial links. Teensyduino is an Arduino add-on to run these sketches.
- **OpenWiNo** (Van Den Bossche et al. 2016): is a free development environment for protocol engineering in WSN and DL-IoT. It permits rapid prototyping of protocols (MAC, NWK and other layers) and pragmatic evaluation of their performances in C language. It also allows them to be run on testbed via real WiNo nodes relying on the Arduino environment and developed in Open Hardware. Coupled with the OpenWiNo software, the WiNo nodes form a self-organized open-protocol mesh network. This approach allows a great versatility on the hardware. Thus, OpenWiNo, associated with the WiNo hardware, has certain advantages compared with the other platforms and testbeds. The main advantage of OpenWiNo is its simplicity. For example, to change the Physical layer on a WiNo node, it is enough to change the transceiver and the associated driver. This is recommended in an open-hardware environment. Different transceivers enabling several PHY layers have been tested successfully with the OpenWiNo: Classical IEEE 802.15.4 2.4GHz DSSS (Freescale), IEEE 802.15.4-2011 UWB (DecaWave DW1000), LoRa mode 868MHz (HopeRF RFM95) and Proprietary 433MHz FSK/GFSK (HopeRF RFM22b).

4.5.2 Scenario 1: Testing with 11 nodes

In this section, we will shed the light on the fact that the proposed contributions are tested on a real-world problem by deploying nodes in the research building of the University Institute of Technology of Blagnac (IUT Blagnac), University of Toulouse, France.

The proposed genetic algorithms are implemented on an Intel core i3-3217U CPU 1.80GHz computer and the following parameters are used. The number of constraints is determined based on our formulation. Length, width and height are determined based on the 3D space of the real experiments:

- Maximum number of generation = 350.
- Number of individuals per generation = 300.
- Probability of mutation = 0.15.
- Probability of crossover = 0.5.
- Number of constraints = 14.
- Length of the area (x) = 23.21 meters.
- Width of the area (y) = 13.95 meters.
- Height of the area (z) = 6.75 meters.
- Maximum execution time = 5400 seconds.

Besides, the theoretical estimated relations (RSSI and FER values) between the fixed nodes are not taken theoretically according to the hardware characteristics of the nodes. Indeed, these relations are based on the average values of a set of practical measures (real ranges). To give those algorithms more ability to reflect the reality when computing the RSSI and FER values,(then better estimating the relations between the fixed nodes and the nomad nodes or between fixed nodes and the mobile node), the corresponding RSSI and FER values are calculated based on the average value of two parameters: a theoretical threshold (we choose between two values of this threshold, according to the existence or not of closed doors between the two concerned nodes) and the measures (RSSI and FER) of the nearest fixed node to the concerned nomad/mobile node.

4.5.2.1 Network architecture

The proposed network is composed of 11 fixed nodes initially deployed, 3 nomad nodes (named 'D', 'E' and 'F'), and a mobile node (named 'C'). The mobile node is attached to a person moving within the 3D space of the building. Figure 4.23 illustrates the power of the RSSI links between the different nodes. The RSSI value of each node is indicated on the arcs (a value between 0 and 255 which is convertible in dBm values (RSSI, 2016) as shown in Figures 4.25, 4.26, 4.27 and 4.28). Blue arcs indicate links with excellent RSSI connections in terms of quality of the received signal, green arcs indicate links with good connections, orange arcs indicate less good connection links, and red arcs indicate weak RSSI connections.

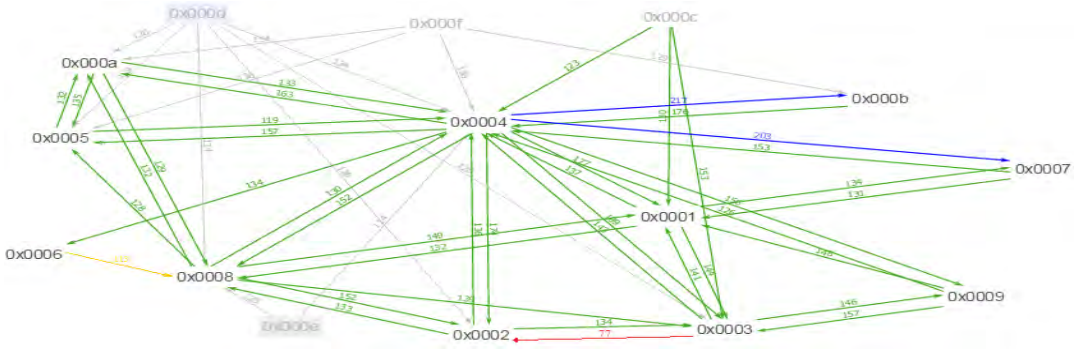


Figure 4.23 The logical network architecture

Although the number of all deployed nodes does not exceeds twenty in our experimentations, the use of meta-heuristics as a resolution approach in our work is justified,

since; according to Cheng et al. (2008); the 3D deployment problem is considered as an NP-hard problem starting from two nodes to deploy. Figure 4.24 illustrates the deployment schema in 2D and 3D plans. The origin of the local taken coordinate system is set at the point $P_0(0,0,0)$ indicated in Figure 4.24(a). The red nodes represent the fixed nodes and the blue nodes represent the nomad added nodes[1]. We note here that our deployment is considered as a 3D deployment (not a 2D multistage deployment) since there is connections between nodes situated in different floors of the building. Besides, the variation in height of the deployed nodes is not negligible as compared to length and width of the deployment field. Consequently, it is recommended to consider the indoor area as a continuous 3D space.

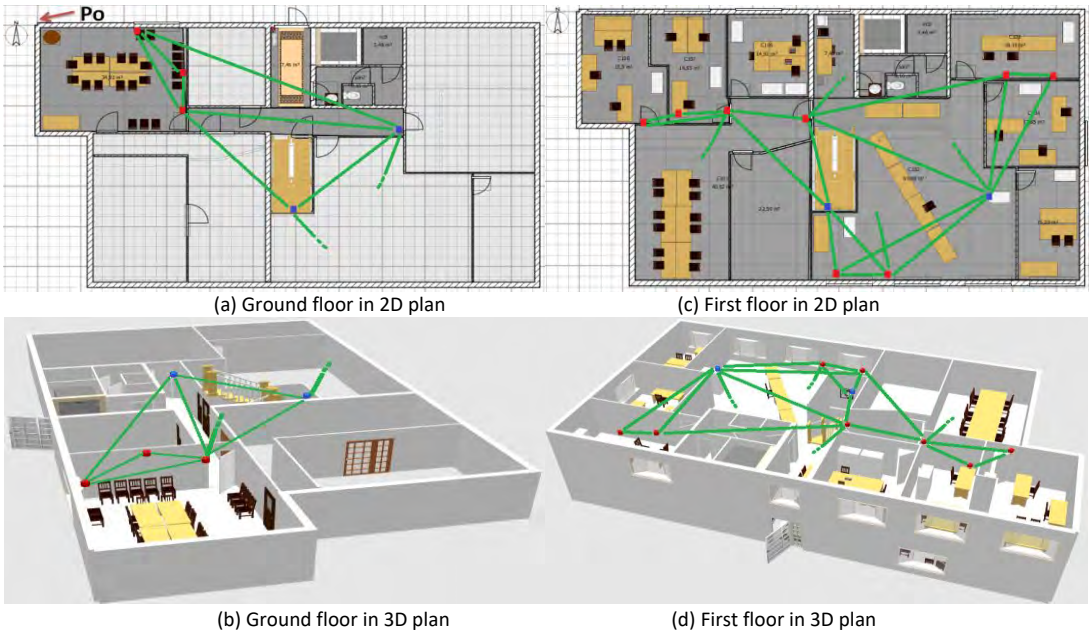


Figure 4.24 The 2D and 3D architecture of the real deployed indoor network

Technical and localization characteristics of the installed nodes are listed in Table 4.38. Table 4.39 illustrates a set of chosen positions taken by the mobile node 'C' on the 3D space to assess coverage and localization before and after the redeployment. These positions are dispersed uniformly in different regions of the 3D space. In both mentioned tables, the x-axis represents the horizontal axis, the y-axis represents the vertical axis, and the z-axis represents the height. The point $P_o(0,0,0)$ corresponds to the following WGS84 GPS coordinates expressed in sexagesimal degrees (in degrees, minutes and seconds): latitude = 43°38'57.4''E; longitude = 1°22'28.4''E and altitude = 164 meters. Theses GPS coordinates can be easily converted into the local coordinates using appropriate formulas.

Table 4.38 Technical and localization characteristics of the installed nodes

N°	Decimal nomenclature	Short address (the node's 16-bit address)	Type	Position according to the local coordinate		
				X	Y	Z
N1	01	0x0001	Teensy 3.0 mk20dx128	278	545	523
N2	02	0x0002	Teensy 3.0 mk20dx128	1063	525	521
N3	03	0x0003	Teensy 3.0 mk20dx128	683	498	526
N4	14	0x0004	Teensy 3.1 mk20dx256	663	414	206
N5	05	0x0005	Teensy 3.0 mk20dx128	2093	305	519
N6	06	0x0006	Teensy 3.0 mk20dx128	1237	1256	443
N7	15	0x0007	Teensy 3.1 mk20dx256	450	00	290
N8	1c*	0x0008	Teensy 3.1 mk20dx256	1114	1252	422
N9	31*	0x0009	Teensy 3.1 mk20dx256	416	495	336
N10	1F*	0x000A	Teensy 3.1 mk20dx256	1813	306	356
N11	34*	0x000B	Teensy 3.1 mk20dx256	672	270	291

[1] Color should be used for this figure in print version

N12	**58**	**0x000D**	**Teensy 3.1 mk20dx256**	**-**	**-**	**-**
N13	**59**	**0x000E**	**Teensy 3.1 mk20dx256**	**-**	**-**	**-**
N14	**60**	**0x000F**	**Teensy 3.1 mk20dx256**	**-**	**-**	**-**
N15	**C**	**0x000C**	**Teensy 3.1 mk20dx256**	**-**	**-**	**-**

Table 4.39 Locations of selected positions taken by the mobile node 'C'

No		**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**P11**	**P12**	**P13**	**P14**	**P15**	**P16**	**P17**
Positions according to the local reference	**X**	943	938	624	345	1152	1393	1814	1646	2148	1904	1748	1167	1693	865	362	1142	2321
	Y	265	422	870	1175	992	1197	1072	435	985	648	25	858	584	520	342	0	0
	Z	392	386	343	518	478	462	394	502	413	517	383	187	10	100	28	140	165

4.5.2.2 Objectives

Our purpose is to add nomad nodes in the indicated locations guaranteeing a set of objectives. These objectives can be either **network objectives** or **application objectives**. The applicative objectives represent the metrics to measure: they are mainly linked to existing sensors measuring physical parameters such as brightness, temperature or opening and closing doors. The network objectives generally concern the measurement of the link's strength between nodes over time: the quality of link (thus the radio coverage quality) is evaluated by measuring the FER, the localization quality is evaluated by measuring the RSSI, while the number of neighbors is evaluated by measuring the both mentioned metrics. OpenWiNo uses the following concept to define a neighbor: a node 'b' is considered as a neighbor in the neighbors table of another node 'a' only if the RSSI signal of 'b', received by 'a' is sufficient (greater than a predefined tunable threshold). In our works we define also a predefined tunable empirical threshold for the FER, below it, a node is not considered as a neighbor. Thus, a neighbor enters to the neighbors table only if the two mentioned thresholds are respected. Indeed, to ensure the 3D coverage, each node must have at least one neighbor and should be monitored by at least one node. In general, to be ensured, the localization requires that all nodes must have at least one neighbor. However, in our case, we use a hybrid 3D localization model based on 3D DV-Hop and RSSI protocol which requires that each node must have four neighbors. We take measures by day and by night. This is explained by the fact that the existence of a large number of people by day implies that the majority of the doors are opened which improve the quality of the received signals. Overnight, the majority of the doors are closed. The anti-fire doors installed in the IUT prevent a good transmission of the signal.

4.5.2.3 Variation of the localization

To measure the localization, a localization model based on RSSI and 3D DV-Hop hybridization is used. Indeed, the localization quality is proportional to the RSSI value. A neighbor may be included in the neighbors table of a node only if it's received RSSI value is greater than the predefined threshold. This threshold is tunable via the command "***phy rssith set***" (Van den Bossche et al 2017). In our case, its value is fixed to 100. Based on the obtained numerical results, we study the effect of the value of the RSSI threshold and its relationship with the FER. The value of the RSSI can change over time and the period of its stability can be less than one second. Given this instability, we take an average value of RSSI extracted from four values taken with an interval of 20 seconds of waiting between these four values; for each pair of nodes (node *i* - node C); $i \in [1,14]$. The RSSI value (noted R_{ci}) that represents the relationship between the mobile node 'C' and each node '*i*' is taken as the maximum value between the detection value of 'C' by '*i*' (signal generated by 'C'), and the detection value of '*i*' by 'C' (signal detected by 'C'). The average of the R_{ci} values between 'C' and all other nodes reaching the fixed threshold in each position Pi is represented by the ordinate axis in Figure 4.25, expressed in the negative value of the dBm (the RSSI values between 0 and 255 are converted (RSSI 2016) into values expressed in dBm), according to the proposed algorithms, by day. The horizontal axis in Figure 4.25 represents the Pi positions indicated in Table 4.39. Figure 4.26 shows the variation of the same RSSI averages overnight. Table A.1 (in Appendix

1) illustrates the average values, in different positions Pi, of the RSSI classified by neighbors of the node 'C', by day. Table A.2 (in Appendix 1) illustrates the average values, in different positions Pi, of the RSSI classified by neighbors of the node 'C', overnight.

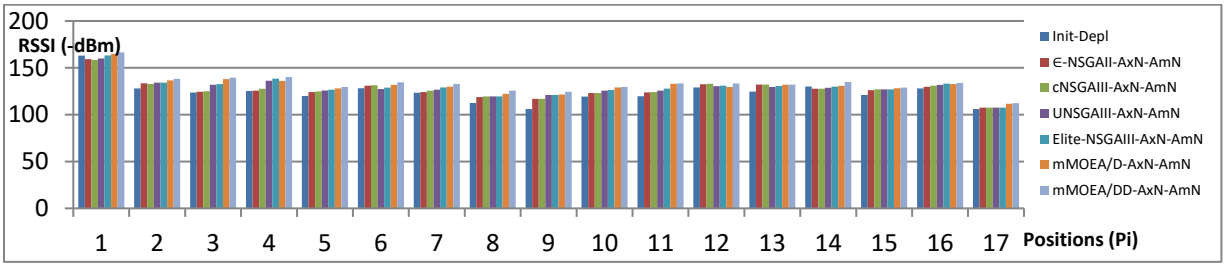


Figure 4.25 Variations on the localization (RSSI), by day, for different positions

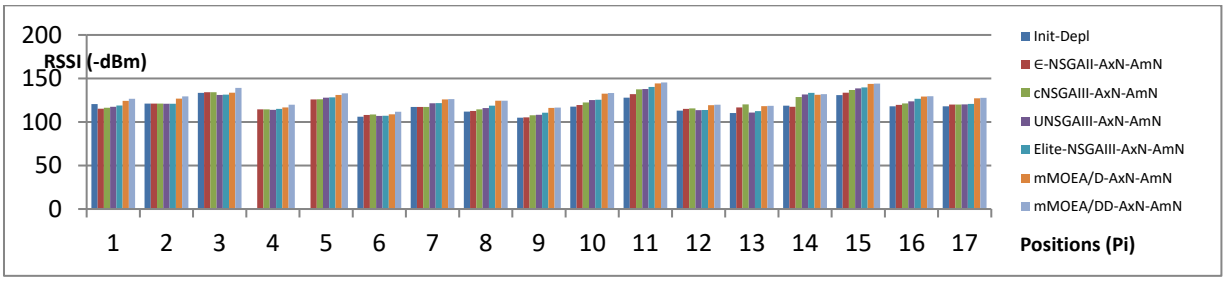


Figure 4.26 Variations on the localization (RSSI), by night, for different positions

4.5.2.4 Variation of the coverage

The frame error rate (FER) is used as a metric to measure the coverage and to evaluate the quality of links between nodes. The above-mentioned threshold of FER (used to introduce neighbors) is fixed to **40%**. The following concept using counters is used to compute the FER: We calculate the couples of exchanged frames (received and lost ones) for each pair of neighbors. These couples are calculated using internal counters which are represented by values between 0 and 255. The use of multiple counters is explained by the fact that if one counter is used, a large number of values can be accumulated, and the historic of the old values can be lost. To calculate the FER for each pair of neighbors, 10 counters of received frames and 10 counters of lost frames are calculated each 10^{-4} seconds. Indeed, in each reception of a new beacon carrying a sequence number noted '*s*', the received frame counter is increased by one and the lost frame counter is increased by a number '*x*' according to the sequence number '*s*'. '*x*' is the difference between the old sequence value and the new received sequence number '*s*'. The sum of the values of lost frames divided by the total amount of frames (those received and lost) gives the FER for the considered couple. The sliding window phenomenon may appear if the first counter is older. The OpenWiNo command "***neighb fer counters***" (Van den Bossche et al 2017) displays the details of internal counters and the sliding windows used to calculate the FER. This principle allows keeping the historic of the exchanged frames and calculating the error rate between each two neighbors. The coverage quality is proportional to the FER value. Despite the FER values vary less than those of the RSSI, we take an average value of the FER extracted from four values taken with an interval of 10 seconds of waiting between these four values; for each pair of nodes (node i - node C); i ∈ [1,14]. The FER value (noted Cov_{ci}) that represents the relationship between the mobile node 'C' and each node '*i*' is taken as the average value between the detection value of 'C' by '*i*' (signal generated by 'C'), and the detection value of 'i' by 'C' (signal detected by 'C'). The average of the Cov_{ci} values between 'C' and all other nodes reaching the fixed threshold in each position Pi is represented by the ordinate axis in Figure 4.27, using the proposed algorithms, by day. The horizontal axis in Figure 4.27 represents the Pi positions indicated in Table 4.39. Figure 4.28 shows the variation of the same FER averages overnight. Table A.3 (in Appendix 1) illustrates the average values, in different positions Pi, of the FER, classified

by neighbors of the node 'C', by day. Table A.4 (in Appendix 1) illustrates the average values, in different positions Pi, of the FER, classified by neighbors of the node 'C', overnight.

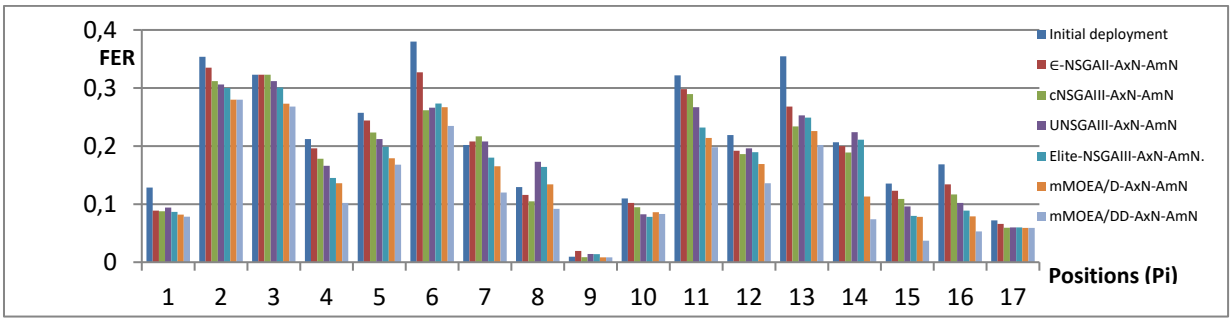


Figure 4.27 Variations of the coverage (FER), by day, in different positions

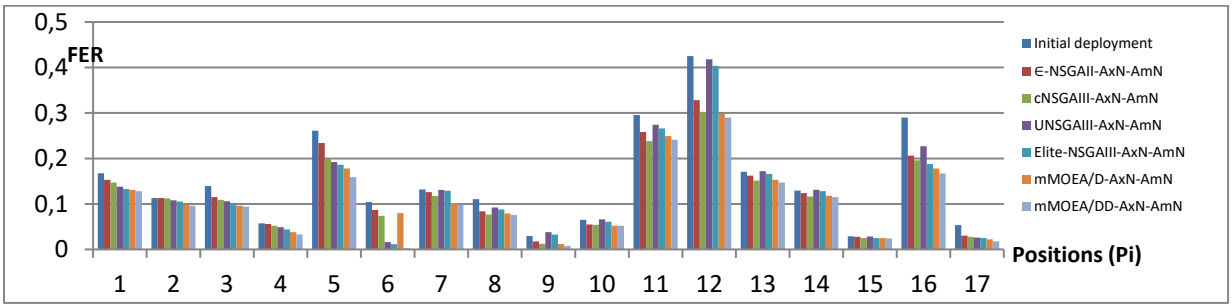


Figure 4.28 Variations of the coverage (FER), overnight, in different positions

4.5.2.5 Variation of the number of neighbors

Figure 4.29 shows the variation of the number of neighbors, using the proposed algorithms, by day, for the Pi positions. Figure 4.30 shows the variation on the number of neighbors overnight, for the Pi positions.

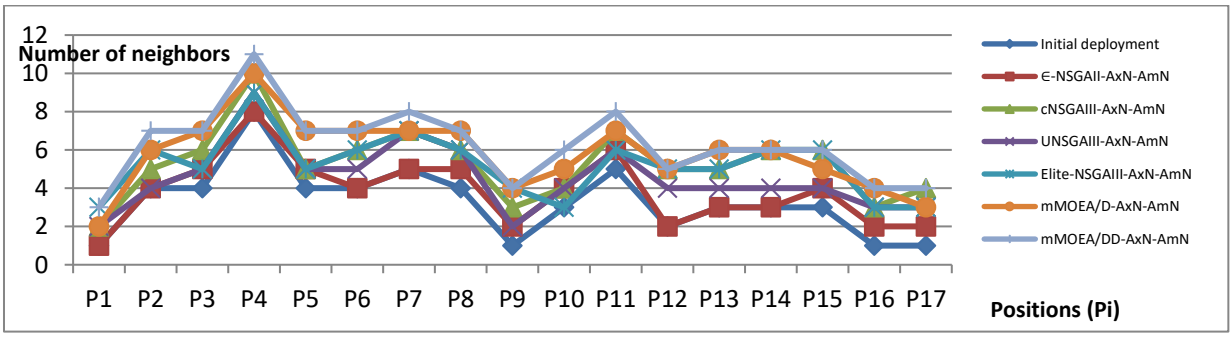


Figure 4.29 Variation of the number of neighbors, by day

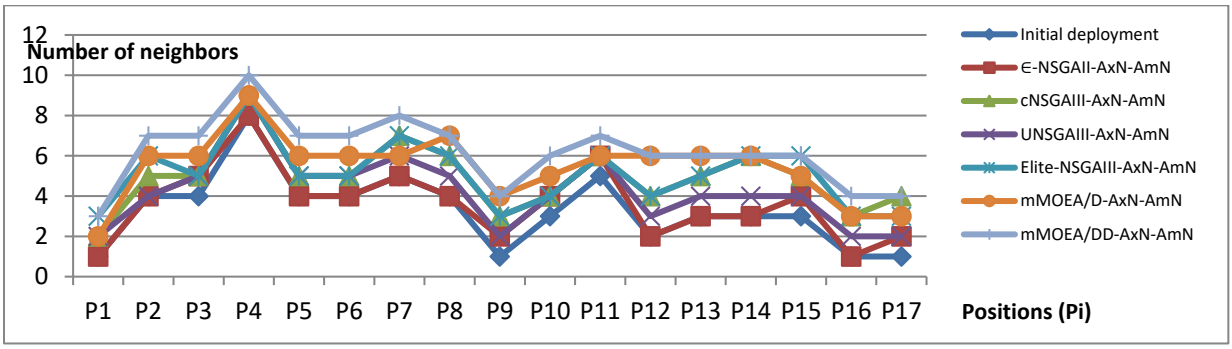


Figure 4.30 Variation of the number of neighbors, by night

4.5.2.6 Discussion

The improvement of the coverage (FER), the localization (RSSI) and the number of neighbors compared to the initial deployment are summarized by day in Table 4.40 and by night Table 4.41.

Table 4.40 The improvement of RSSI and FER rates compared to the initial deployment, by day

\	NSGA-IIr	NSGA-IIa	cNSGA-IIIr	cNSGA-IIIa	U-NSGA-IIIr	U-NSGA-IIIa
Average RSSI's improvement	+3,0929 (+1,2081%)	+3,5258 (+1,3772%)	+4,7547 (+1,8573%)	+5,8252 (+2,2754%)	+7,5488 (+2,9487%)	+9,5194 (+3,7185%)
Average FER's improvement	-2,019%	-3,463%	-3,247%	-4,307%	-6,087%	-8,175%
Average number of added neighbors in all Pi positions	+0,5294	+1,6071	+1,375	+1,353	+1,5076	+2,94

Table 4.41 The improvement of RSSI and FER rates compared to the initial deployment, by night

\	NSGA-IIr	NSGA-IIa	cNSGA-IIIr	cNSGA-IIIa	U-NSGA-IIIr	U-NSGA-IIIa
Average RSSI's improvement	+1,627 (+0,6355%)	+3,774 (+1,4742%)	+4,205 (+1,6425%)	+4,727 (+1,8464%)	+8,7211 (+3,4066%)	+10,132 (+3,9578%)
Average FER's improvement	-2,018%	-3,463%	-3,247%	-4,307%	-6,087%	-8,1754%
Average number of neighbors in all the Pi positions	+0,3529	+1,647	+0,9411	+1,7647	+2,1176	+2,8823

After the analysis of the experiment results, several findings can be considered:

- In the majority of the instances, the average of RSSI signal strength is greater at day than night because of the opened doors during day while the average of FER signal strength is greater at night than day due to less human activities at night involving less perturbations and signal interference.
- When comparing the variation of the FER and the RSSI rates between day and night, it is noted that the FER rate is higher by day than night although the RSSI rate is also higher by day than night: this indicates that the introduction of neighbors according to the highest RSSI does not always give the lowest error rate. By comparing the error rate before and after deployment, we conclude that our approach has allowed minimizing the error rates and subsequently maximizing coverage.
- The node range is not spherical: according to measures, some nodes can be detected by a set of nodes while some other further near nodes cannot detect them. For instance, the node 'N4' is the only one detecting the node 'C' which is in the location P1 although there is other nodes which are less distant from the location P1. This assumption has been considered when implementing the proposed algorithms.
- When capturing measures on the same position over several periods, it is noted that after a period of initializing the links between nodes, the RSSI increases (stronger signal) and the FER decreases (less error rate).
- The nature of the relationship between the RSSI and the FER is investigated: each neighbor is introduced in the neighbors table based on a high RSSI rate but after a moment, the rate of lost frames may be very high. Thus, increasing the RSSI value may not decrease the FER value. So, the FER indicates the quality of links better than the RSSI.
- When comparing the two methods of selecting the mutation operators (random vs adaptive), we conclude that the adaptive method is always better than the random one due to its intelligent process. Thus, the adaptive method can be considered as an upper bound for the random method.
- According to experimentations, the U-NSGA-III proposed versions are more efficient in resolving our problem than the NSGA-II proposed versions. This can be explained by the fact that the NSGA-III selection procedure which is based on reference points and niching (U-NSGA-III) allows more diversity among the members of the population.
- Although this selection procedure used on the cNSGA-III (based on reference points), the NSGA-II seems to be (in several positions) more efficient than the cNSGA-III: this can be

explained by the fact that the cNSGA-III is dedicated to resolve many objective problems and may have some difficulties when resolving single or bi-objective problems.

4.5.3 Scenario 2: Ophelia testbed (Testing with 36 nodes)

Ophelia is a platform based on Arduino, OpenWiNo and the installed TeensyWiNo nodes, to which a web interface is added. This interface allows remote access to Ophelia testbed, programming and executing experimental sketches (in python language) on the nodes. In what follows, we detail the execution scenario of our experiments using Ophelia.

4.5.3.1 Experimental scenario and results

30 fixed nodes initially deployed with known positions (chosen according to the users application needs) are used. The number of nomad nodes to be added is limited to six. The positions of the latter nodes are to be determined by the tested optimization algorithms. Only one mobile node is applied.

In order to study the impact of the selected positions of the nomad nodes on the network performance, the following scenario of the experiments is repeated several times: At the beginning, all the nodes are flashed and the parameters of the initial configuration (power of transmission, etc.) are sent. Then, the mobile node sends a broadcast to all other nodes. The RSSI and FER rates issued from and received by each node are considered. After a predefined waiting time, another transmitter is chosen and the other nodes become receivers. This process is repeated until 36 experiments are achieved. At the end, two connectivity matrices, combining the RSSI and FER means between the nodes, are created. The average number of neighbors of each node is deduced from these two matrices. In our experiments, two nodes are considered neighbors if and only if the mean rate of the RSSI (and FER, respectively) recorded between these two nodes is greater (lower, respectively) than a pre-defined threshold equal to 100 (0.1, respectively). Due to the stochastic nature of the used evolutionary optimization algorithms, the use of a statistical test with several executions is necessary to assess their behavior. Therefore, the mean values in our tests are calculated based on 25 runs of the algorithms.

Comparing the RSSI values: To assess different objectives, like the localization, the connectivity or the quality of links, the RSSI metric is calculated. Our experiments are based on a hybrid localization model which combines the RSSI information and the Distance-VectorHop protocol. Indeed, the higher the RSSI value, the better the localization will be. Figure 4.31 shows the average RSSI values (a convertible to dBm value in [0, 256]) exchanged between nodes for different numbers of objectives.

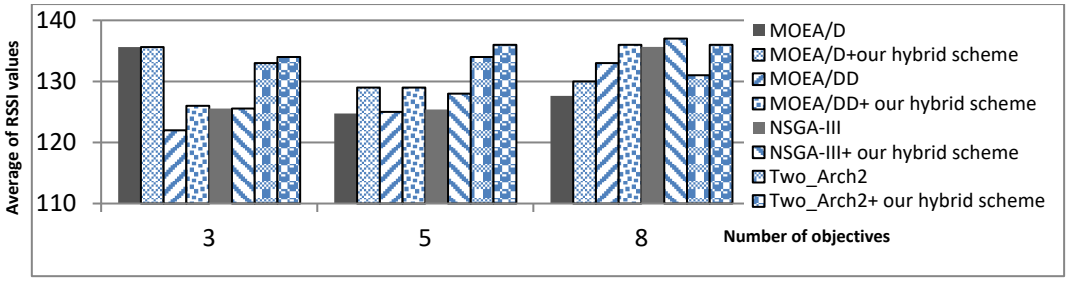


Figure 4.31 The average RSSI values, for different number of objectives

Comparing the FER rates: To assess the coverage and the quality of links between nodes, the FER metric is calculated. Indeed, the lower the FER value, the better the coverage will be. To assess FER values for each pair of nodes, an average value deduced from four values is considered with an interval of 10 seconds between the four values. Figure 4.32 shows the average FER values between nodes for different numbers of objectives.

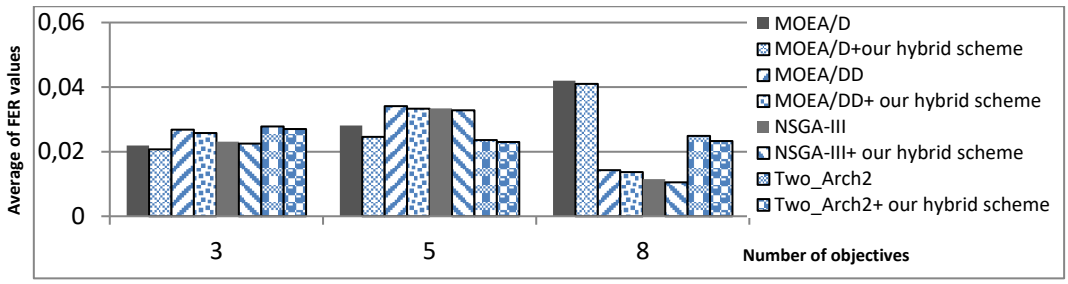


Figure 4.32 The average FER values, for different number of objectives

Comparing the number of neighbors: To evaluate the connectivity and the utilization of the network, the average number of each node neighbors is computed. Indeed, the previous definition of the neighbor based-on RSSI and FER is used. Figure 4.33 shows the average number of neighbors for all nodes and for different numbers of objectives.

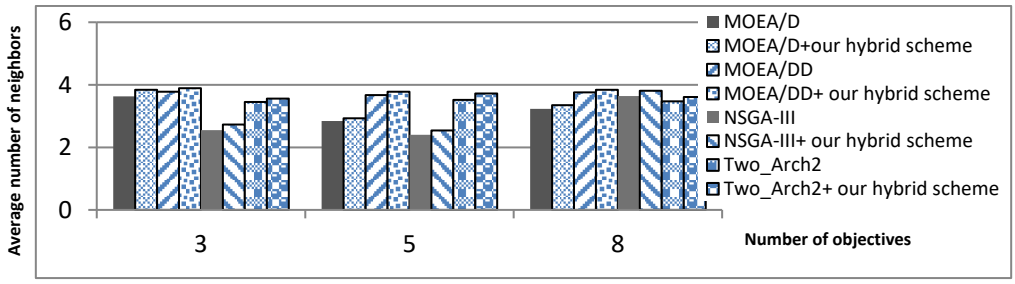


Figure 4.33 Average number of neighbors, for different number of objectives

Comparing the network lifetime: Figure 4.34 indicates, for different number of objectives, the network lifetime considered as the time in which the first node of the network has no energy.

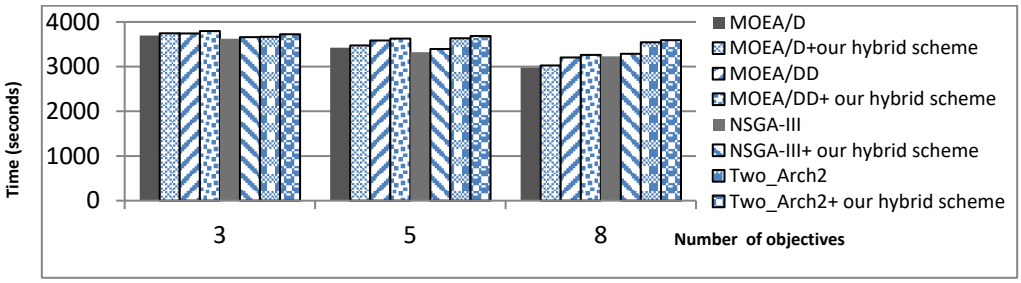


Figure 4.34 Average lifetime of the network, for different number of objectives

Comparing the energy consumption: Figure 4.35 illustrates the variation of the network energy consumption as a function of time. Indeed, the average of the energy rates of all nodes is calculated after adding the nomad nodes.

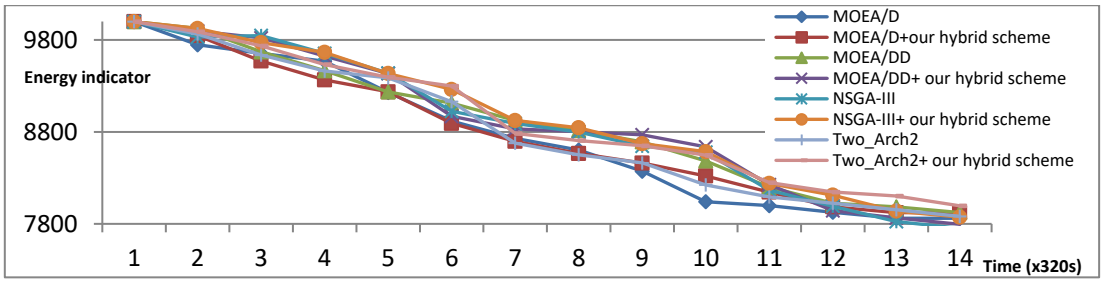


Figure 4.35 Average energy consumption levels, as a function of time

4.5.3.2 Interpretations and discussion

After performing the experiments, several findings can be considered:

- This study is a proof of the applicability of the optimization approaches in real-world contexts using real experiments. It shows the theoretical findings obtained by applying the tested algorithms.
- Our experiments show that a link between two nodes can have a high FER and an excellent RSSI at the same time. Thus, the RSSI and FER rates are not always inversely proportional.
- By studying the performance and behavior of the tested algorithms before and after the application of the proposed hybridization scheme, the experimental results demonstrate that NSGA-III is generally better than MOEA/DD in the RSSI and FER rates. Thus, the NSGA-III is considered more efficient in satisfying the localization, the coverage and the link quality, while MOEA/DD is better used to satisfy the average number of neighbors and the network lifetime.
- Consistent with our numerical results (see sections 4.2 and 4.3), the experiments assert that the effectiveness of the algorithms depends on the number of objectives to be optimized. Indeed, figures 4.31, 4.32, 4.33 and 4.34 show that if the objective number does not exceed three, the behavior of the MOEA/D will be better than that of the NSGA-III. In the case of four objectives or more, the behavior of the NSGA-III becomes better than that of the MOEA/D. This finding can be explained by the fact that, unlike MOEA/DD and Two_Arch2, the NSGA-III is only dedicated to many-objective problems.
- Contrary to different studies such as (Li et al. 2015) showing that the decomposition-based methods are generally more efficient than the NSGA-III, our results prove that MOEA/DD is not always better than NSGA-III because our problem is a real-world one having some features which are different from those characterizing the theoretical problems used to evaluate these algorithms.
- Finally, it is proven that the incorporation of the dimensionality reduction and the user preferences improves the results (higher HV, higher coverage and localization) and enhance the behavior of the tested algorithms.
- The Two_Arch2 has an almost stable behavior that is generally not influenced by varying the number of objectives.

4.5.4 Results of experimentations on PI-NSGA-III

Comparing the RSSI rates*:* To assess the connectivity (*f8* in our modeling, see section *3.2.2*), the quality of links (*f10*) and the localization (*f5*), the RSSI metric is used since the used localization model is based on RSSI and Distance-VectorHop protocol. Thus, the higher the RSSI rate is, the better the localization is. Figure 4.36 illustrates the average of RSSI (a value in [0, 256] convertible to dBm) between nodes (for each node i ∈ [1, 36] with other ones).

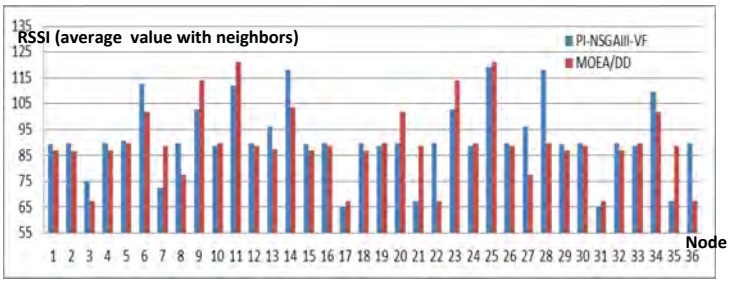


Figure 4.36 Comparing the average rates of RSSI

Comparing the FER rates: To measure the coverage (*f6*), we use the FER as a metric to evaluate the quality of links between nodes. Hence, the lower the FER is, the better the coverage is. Figure 4.37 shows the average measured FER for each node with other ones.

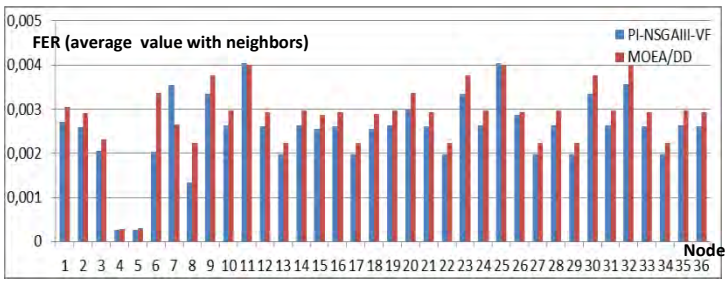


Figure 4.37 Comparing the average rates of FER

***Comparing the number of neighbors**:* To measure the network connectivity (*f8*) and the utilization of the network (*f4*), the average number of *neighbors* of each node is measured. The same definition of neighborhood (previously explained in the experiment scenario) is used. Figure 4.38 illustrates the average number of neighbors for each node.

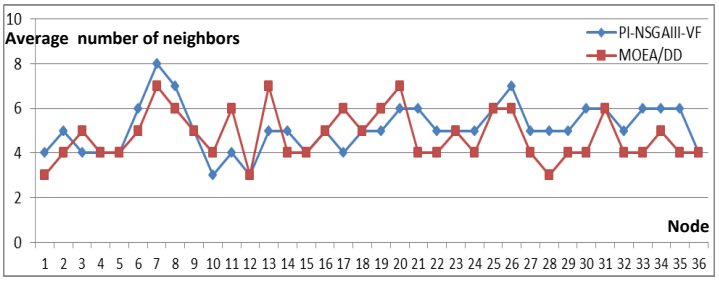


Figure 4.38 Comparing the average number of neighbors

Discussion and interpretations of PI-NSGA-III results: After evaluating the experiments, different interpretations can be deduced:
- The rates of RSSI and FER are not necessarily inversely proportional: a link between two nodes may have an excellent RSSI and high FER in the same time.
- The FER values are more relevant during day then overnight. This can be explained by the activities of persons in the building during day such as opening and closing doors, which generates perturbations on the signal.
- By comparing the efficiency of the tested algorithms, the experimental results show that the NSGA-III generally outperforms the MOEA/DD in the RSSI and FER rates (Thus, it is better in satisfying the coverage, the quality of links and the localization), while the latter algorithm is generally more efficient than NSGA-III in optimizing the number of neighbors of each node (Thus, it is better in satisfying the utilization of the network).
- Contrary to the results of different studies (see (Li et al 2015) and (Yuan et al 2016)) showing that the decomposition-based methods are better than NSGA-III, our results prove that the MOEA/DD does not always outperforms the NSGA-III (PI-NSGA-III-VF in our case). This can be explained by the fact that our problem is a real world one which is different from theoretical problems used to test the algorithms. It is also explained and by the contribution of the incorporation of preferences on the NSGA-III.
- Since the tested algorithms (NSGA-III and MOEA/DD) are tested by their authors only on instances of theoretical test problems, our study is a proof of the applicability of the optimization approaches in real world contexts using real experimentations.

4.6 Results on hybridizing ACO and NSGA-III (AcNSGA-III)

4.6.1 Numerical results of the algorithms

When resolving a particular problem, the parameters setting has a major influence on the algorithm behavior. In this section, we present the parameters setting of the algorithms and the results using the HV metric. The used parameters are as follows:

- The size of the population (***NS***): 300 (Thus, ***Npop*** =150).
- The operators of reproduction: It would be best to perform a recombination operation using near parent solutions in the case of MaOPs. Therefore, a large distribution index SBX operator (simulated binary crossover) is used. The recombination probability is PrOx=0.8 using a distribution index τ_r = 45. The mutation probability using the bit-flip operator, is PrMt=$2*10^{-3}$ using a distribution index τ_m = 25.
- The number of runs: in order to guarantee statistical confident results, each algorithm is executed 25 runs using a different initial population in each execution.
- The constraints number: 5.
- The number of ants (***NbA***): 350
- The pheromone minimum threshold (***MaxTP***): 1
- The pheromone maximum threshold (***MinTP***): 15
- The pheromone exponent (***ExP***): 0.4
- The pheromone evaporation coefficient (***EvP***): 0.25
- The number of objectives and the termination condition (the maximum number of generations) are as shown in Table 4.42. For each instance, the best performance is demonstrated using shaded background.

Table 4.42 HV (Best, average and worst values)

Obj Nbr	Max nbr of generations	ACO	NSGA-III	AcNGSA-III
3	**1300**	0.984682	0.902458	0.903168
		0.983561	0.901896	0.902375
		0.982327	0.898023	0.987653
4	**1800**	0.885236	0.974685	0.976687
		0.884381	0.974233	0.975986
		0.884003	0.973612	0.975124
5	**2600**	0.878847	0.972892	0.973382
		0.872324	0.972716	0.972899
		0.871456	0.972684	0.972633

The AcNGSA-III outperforms the ACO and the NGSA-III in the most of the instances which proves its efficiency. Another observation is that the ACO is better than the NSGA-III for three objectives, while the latter one is better when the number of objectives is higher than three. This is is congruent with the observation of its authors (Deb and Jain 2014) which asserts that the NSGA-III is dedicated to resolve many-objectives problems. Also, due to the increasing complexity of the problem when the number of objectives increases, for all algorithms, better HV values are obtained with smaller number of objectives.

4.6.2 Comparing simulations and experimentations

This section aims to provide a comparison between the simulation and the experimental tests of the 3D deployment scheme in indoor WSN while satisfying different objectives. We are interesting in testing the behavior the tested evolutionary optimization algorithms (ACO, NSGA-III) and the new suggested one (AcNSGA-III). The same above mentioned simulation/experiments parameters and working environment are used.

Experimental/simulation scenario: In experiments**,** 30 fixed nodes with known positions are used. Six nomad nodes are to be added. Their positions are determined with the tested optimization algorithms. For mobile nodes, only one node is used. The positions of the initially deployed fixed nodes are chosen according to the users applicative needs. The execution scenario is as following: At the beginning, all the nodes are flashed. Then, the initial configuration parameters (transmission power, etc.) are sent. Then, a first node sends a broadcast to other nodes. The measurements are taken in two directions: the sender records his RSSI and FER rates with each node and each receiving node returns these same measurements. After a fixed waiting time, the sender finishes the process. Then, the sender is

changed and all other nodes become receivers. The same process is repeated until 36 experiments are performed which give us two connectivity matrices with the RSSI and FER values between all nodes. The neighbors of each node are deduced from the two matrices. In our case, a node *i* is considered as neighbor to an another node *j* if the average of the RSSI emitted from *i* to *j* and the onefrom*j* to *i*, is greater than a fixed threshold (100); and the FER average is also lower than a fixed threshold (0.1).

To study the impact of the new positions of the nomad nodes on the network performance and to evaluate the behaviors of the suggested algorithm (acNSGA-III), this latter is compared to the ACO and the NSGA-III (Deb and Jain 2014) which is another recent many-objective optimization algorithm. Because of the stochastic nature of evolutionary algorithms and the necessity of a statistical test to compare two algorithms, the average values in the experimental scenario are obtained using 25 executions. Thus, we obtain a well-based judgment concerning the performance of the algorithms.

For simulations, the same architecture (number and types of nodes) and scenario are used. Concerning the fixed nodes, its positions are identified using the distribution law of the OMNeT++. This distribution law aims to distribute nodes starting from the center of the RoI uniformly. Concerning the connectivity of nodes in simulations, a connectivity matrix based on empirical experiments is established between nodes using the same initial connectivity links of the experiments.To introduce dynamism on the network in simulations; the RSSI connectivity links are set to perturbations to modify the initial links. Indeed, the RSSI matrix is set to a perturbation (+/- 30 for each value) to have new connectivity relations between nodes each time. The simulation scenario is as follows: An initial message is sent from the trigger node (mobile node) to a random destination ***d***. Once ***d*** is found by the AODV routing protocol, it becomes the source and a new destination is selected until the maximum time of simulation is reached.

Since comparing two algorithms is only possible through the use of a statistical test and due to the stochastic nature of the tested evolutionary algorithms, it is necessary to perform any test over many executions to obtain a well based judgment concerning their performance. Thus, all average values in this section, are computed based on 25 executions of the algorithms. In the following, the different measures (RSSI, FER, etc.) are taken in both simulation and experiments.

Comparing the RSSI rates. In order to measure the localization, the RSSI metric is used since the used localization model is based on hybridization between the RSSI and the DVHop (Distance-VectorHop) localization protocol. Thus, the higher the RSSI, the better the localization is. A neighbor can be added to the neighborhood table of a node only if the RSSI value of the detected node is greater than a predefined threshold. The value of this threshold is set to 90. Initially, RSSI levels are based on our empirical experiments. Then, as mentioned before, in order to guarantee dynamism within the network, disruption of the value of RSSI is introduced via a random function. Figure 4.39 shows, for different number of objectives, the average of the RSSI rates measured for all the nodes in connection with the mobile node.

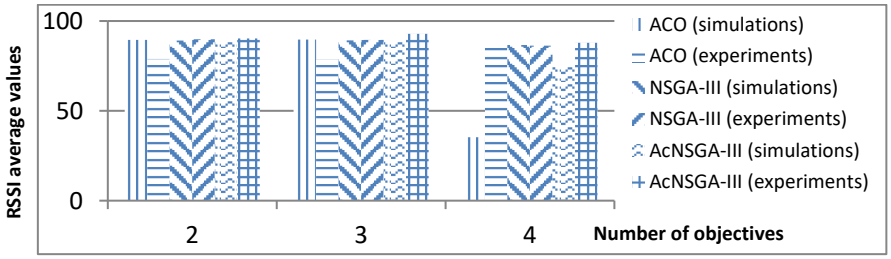


Figure 4.39 Comparing the average RSSI rates

Comparing the FER rates. To measure coverage, we use the FER as a metric to assess the quality of links between nodes. Thus, the lower the FER, the better the coverage is.

Although the FER values are less variable than those of the RSSI, for each node's pair {node C , node i}, an average value extracted from four values is taken with an interval of 10 seconds between the four values. Initially, FER rates are based on our empirical experiments. Afterwards, to guarantee dynamism within the network, disruption of the FER values is introduced via a random function (+/- 0.04 to +/- 0.2). Figure 4.40 shows the average FER rates measured for all nodes in connection with (detected by / detecting) the mobile node.

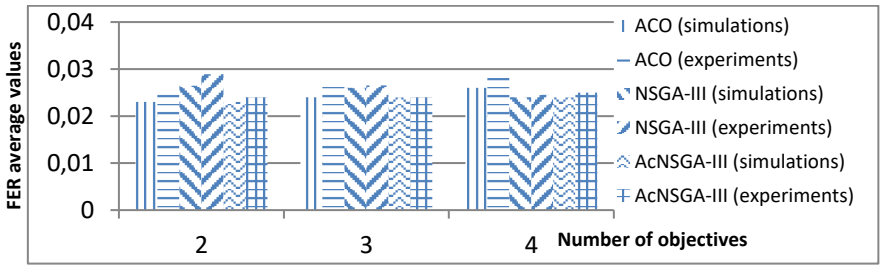


Figure 4.40 Comparing the average FER rates

Comparing the number of neighbors. In order to assess the network connectivity, the number of neighbors of the target (the mobile node) is measured. Figure 4.41 shows the average of the number of neighbors of the mobile node per objective.

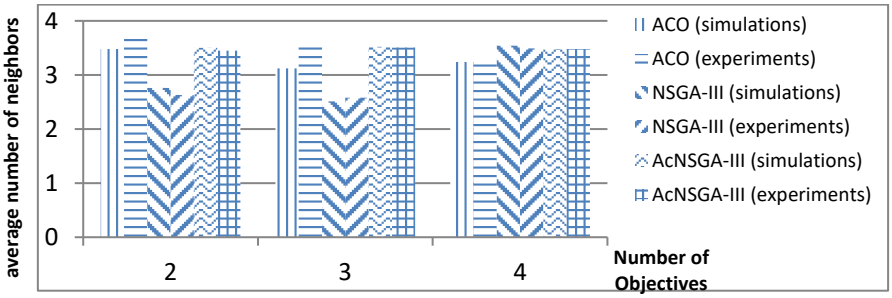


Figure 4.41 Comparing the average number of neighbors

Comparing the energy consumption and the network lifetime. Figure 4.42 shows the variation of the energy level of the network according to the time. Indeed, for the different tested algorithms, according to the number of the fixed nodes, an average of the energy indicator of all nodes of the network is measured after adding nomad nodes.

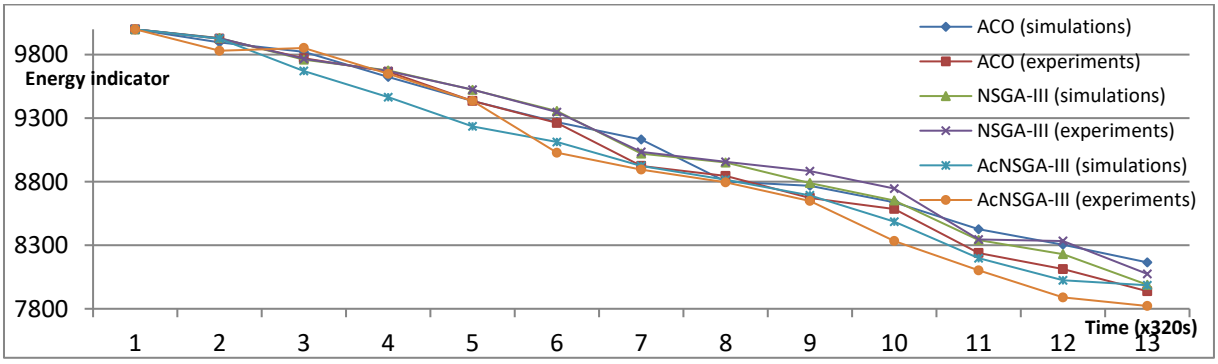


Figure 4.42 Comparing the average energy consumption levels

Figure 4.43 illustrates the lifetime of the network. It shows for different number of objectives, the time in which the first node of the network is switched off.

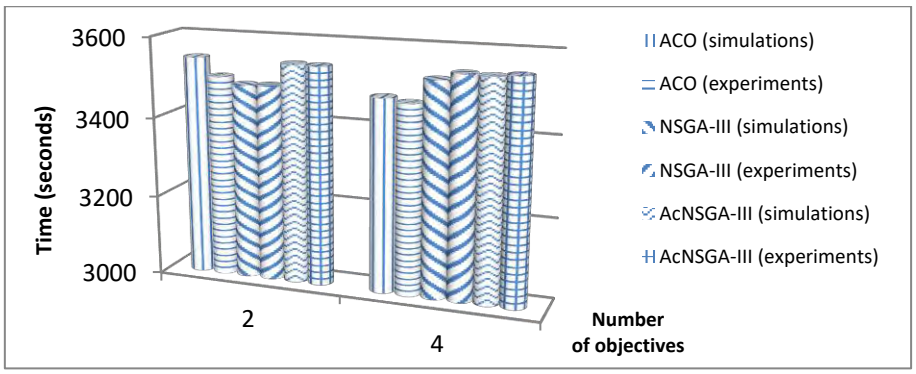


Figure 4.43 Comparing the average lifetime

Discussion and interpretations

After comparing the simulations and the experiments, different interpretations can be considered:

- The obtained results (Figures 4.39 to 4.43) show conformity with the results of experimentation, notably with regard to the coverage and localization rates. This proves the accuracy of the models of simulation and the effectiveness of the proposed approach in different contexts. Indeed, our work represents a proof by experimentation and simulations of the observations which has been proved by the authors of the tested algorithms (NSGA-III for example) only by tests on instances of theoretical test problems.
- In several cases, lower RSSI averages are recorded after adding the nomad nodes. Despite this decrease indicating that the RSSI rates of the added nodes are lower than the RSSI values of the fixed nodes, the localization rate, the coverage rate and the number of neighbors are improved. Given the objectives set by our approach, this decrease in RSSI averages is understandable, since adding a node in a location x1 so that it will be close to several nodes with a lower RSSI value will be better than adding it in a location x2 with a higher RSSI value but smaller number of neighbors.
- The error rates (FER) are more important in experiments then in simulations. This is due to the influence of the activities of persons in the building during experiments (for example opening and closing doors) which generates the perturbation of the signal.
- By comparing the efficiency of the tested algorithms, as proved by numerical results (section 4.3), the simulation/experimental results show that this efficiency is relative to the number of objectives to be optimized. Table 4.42 shows that for less than three objectives, the ACO is more efficient than NSGA-III, while the latter is more effective than ACO if the number of objectives exceeds three. This is explained by the fact that the ACO is dedicated for multi-objective problems, while the authors of NSGA-III propose this latter as an adaptation of the NSGA-II for many-objective problems having more than three objectives. However, the AcNSGA-III has an almost stable behavior and is not influenced by the variation of the number of objectives.

4.7 Results on accentBirdsPSO and on hybridizing PSO and MAS

4.7.1 Numerical results

In this section, the algorithms parameters and the obtained numerical results are presented.

4.7.1.1 PSO and NSGA-III parameters

The used setting parameters are detailed in Table 4.43.

Table 4.43 Setting parameters of the algorithm

Parameter	Value
Population size	300
Number of independent runs	25, on different initial populations
Number of constraints	7

Number of objectives			Variable, see Table 4.44
Maximum number of generations			Variable, see Table 4.44
PSO parameters	Number of particles per swarm		10-50
	Number of swarms		Identified by the used clustering algorithm
	C1(cognitive components)		2.8 to 2.2
	C2 (social components)		1.2 to 1.8
	Inertial weight		0.95 to 0.4
	Maximum particle velocity		According to the formula (5.1)
	Initial swarm particle velocity		Randomly uniformly distributed in [-4, 4]
	Initial minimum number of clusters		4
NSGA-III parameters	Recombination	Operator	SBX
		probability	0.8
		distribution index	45
	Mutation	Operator	Bit-flip
		probability	1/400
		distribution index	25

The sum of cognitive and personal components is generally set to four as recommended in (Shi and Eberhart 1999). In (Jason et al., 2004), they are set to 2.8 and 1.3, respectively. Although, based on our experiments, we adopt the values of 1.75 and 1.35, respectively. According to (Shi and Eberhart 1999), the linear inertia weight value decreasing from 0.95 to 0.4.If the maximum velocity of the particle is too low, the algorithm becomes too slow; else if it is too high, the algorithm becomes too unstable. Thus, the maximum particle velocity is set according to the formula **5-4*(*i*/*I*)** (5.1) where *i* is the current iteration and *I* is the maximum number of iterations.

4.7.1.2 Results on our real-world problem

Given the random behavior of the optimization algorithms and aiming at obtaining a statistically reliable comparison results, an average of 25 executions of the algorithms is performed. Table 4.44 demonstrates the obtained HV values for different numbers of objectives and generations. Best performances (having higher HV) are shown with shaded backgrounds.

Table 4.44 Hypervolume values (Best, average and worst)

ObjNbr	Max nbr of generations	NSGA-III	MaOPSO	acMaPSO	acMaMaPSO
3	1300	0.902231	0.903458	0.903631	0.903774
		0.901658	0.902896	0.903036	0.903039
		0.898235	0.898023	0.902563	0.901088
4	1800	0.974892	0.976985	0.977331	0.977329
		0.974743	0.976833	0.977098	0.977243
		0.973897	0.975612	0.976892	0.976838
5	2600	0.972983	0.972892	0.972985	0.973324
		0.972563	0.972116	0.972728	0.972984
		0.972126	0.971084	0.972436	0.972356

Obtained results show that for different numbers of objectives, acMaPSO and acMaMaPSO are often more efficient than other algorithms. Moreover, although the MaOPSO is more efficient than the NSGA-III, it has higher relative degradation when the number of objectives increases compared to other algorithms.

4.7.1.3 Results of DTLZ test problems

To prove its scalability, we test the performance of our approach on the theoretical test suite (DTLZ **(Deb et al 2005))** after assessing it on real testbeds. The quality of the obtained non-dominated solution sets generated by the tested algorithms is assessed using the IGD metric **(Coello et al 2002)** and 15 independent runs in each test. IGD allows measuring both the diversity and the convergence of the non-dominated obtained solutions. Table 4.45 shows best, average and worst IGD values of the DTLZ1-4 test suite. Best performances (having lower IGD values) are shown with shaded backgrounds.

Table 4.45 Best, average and worst IGD values onDTLZ1-4 test suite

Test Pb	ObjNbr	Number of generations	MaOPSO	NSGA-III	acMaPSO	acMaMaPSO
DTLZ1	3	400	3.458E-4 1.231 E-3 4.783 E-3	4.880 E-4 1.308 E-3 4.880 E-3	3.566 E-4 1.209E-3 3.844E-3	3.238E-4 1.298E-3 4.257E-3
	5	600	0.635E-4 0.684E-4 1.323E-3	5.116 E-4 0.980 E-4 1.979 E-3	0.642E-4 0.649E-4 1.589E-3	0.637E-4 0.648E-4 1.635E-3
DTLZ2	3	250	5.663 E-4 6.102 E-4 7.029 E-4	1.262 E-3 1.357 E-3 2.114 E-3	3.856 E-4 5.897 E-4 8.014 E-4	5.734 E-4 5.023 E-4 6.298 E-4
	5	350	1.302 E-3 1.534 E-3 1.896 E-3	4.254 E-3 4.982 E-3 5.862 E-3	1.279 E-3 1.432 E-3 1.982 E-3	1.243 E-3 1.412 E-3 1.786 E-3
DTLZ3	3	1000	7.235 E-4 3.028 E-3 6.657 E-3	9.751 E-4 4.007 E-3 6.665 E-3	6.088 E-4 3.011 E-3 6.234 E-3	6.320 E-4 2.875 E-3 4.657 E-3
	5	1000	1.432 E-3 2.569 E-3 6.233 E-2	3.086 E-3 5.960 E-3 1.196 E-2	1.533 E-3 2.167 E-3 5.798 E-3	1.512 E-3 2.247 E-3 6.064 E-3
DTLZ4	3	600	2.846 E-1 6.280 E-1 6.685 E-1	2.915 E-4 5.970 E-4 4.286 E-1	4.321 E-1 7.194 E-1 8.632 E-1	3.654 E-1 6.025 E-1 6.309 E-1
	5	1000	3.093 E-1 4.834 E-1 4.992 E-1	9.849 E-4 1.255 E-3 1.721 E-3	6.058 E-2 0.381 E-1 2.984 E-1	8.052 E-2 0.352 E-1 2.554 E-1

For DTLZ1-3, the same results as our real word problem are found: acMaPSO and acMaMaPSO are more efficient than other algorithms while MaOPSO is better than NSGA-III. For DTLZ4, NSGA-III is more efficient than acMaPSO, acMaMaPSO and MaOPSO. There is no significant difference between the results of three and five objectives.

4.7.2 Experimental and simulation results

In this section, a comparison between the simulation and the experimental tests is provided. The performance and the behavior of the suggested algorithms (acMaPSO, acMaMaPSO) are compared to those of MaOPSO and NSGA-III. Indeed, theoretical analysis and simulators are not entirely able to reproduce all the technical and physical characteristics of the real environment. Also, nowadays, there is a tendency to face protocol and algorithmic proposals with real environments. In this context, through the experiments over our testbed, we aim to reduce the gaps between theory and practice in WSN and IoT deployment issues.

4.7.2.3 Comparing the simulations to the experimental results on Ophelia testbeds

*** Experimental/simulation scenario**

The same scenario in the section 4.6.2 is used, with average values of 25 executions of the algorithms.

*** Comparing the RSSI rates:** To assess the connectivity (*f8* in our modeling (see section 3.2.2)), the quality of links (*f10*) and the localization (*f5*), the RSSI metric is used since localization is based on RSSI and Distance-Vector Hop protocol. So, the higher the RSSI rate, the better the localization. Figure 4.44 illustrates, for different numbers of objectives (to be satisfied by the tested algorithms), the average of the RSSI rates measured for all nodes in connection with (detected by/detecting) the mobile node. This average of RSSI is a value in [0, 256] convertible to dBm.

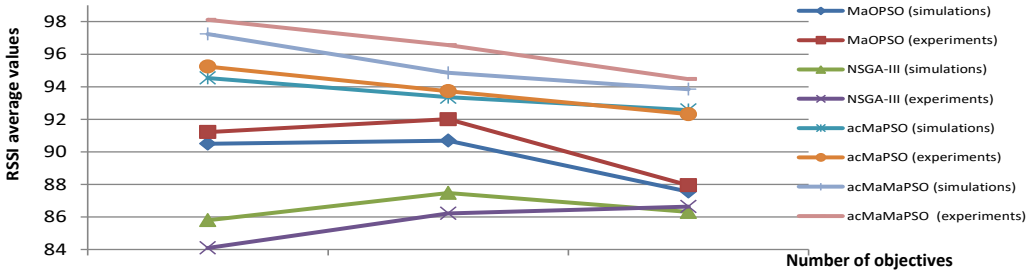


Figure 4.44 RSSI average rates of nodes in connection with the mobile node

*** Comparing the FER rates:** To measure the coverage (*f6*), FER is used as a metric to evaluate the quality of links between nodes. Thus, the lower the FER is, the better the coverage is. Figure 4.45 illustrates, for different numbers of objectives to satisfy by the tested algorithms, the average of the FER rates measured for all nodes in connection with (detected by/detecting) the mobile node.

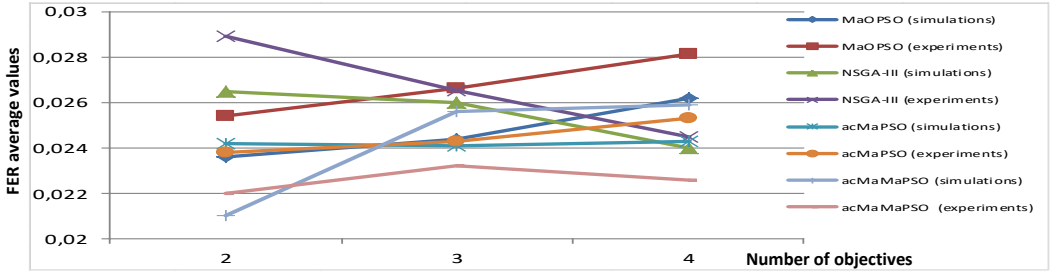


Figure 4.45 FER average rates of nodes in connection with the mobile node

*** Comparing the number of neighbors:** To measure the network connectivity (*f8*) and the utilization of the network (*f4*), the average number of neighbors of nodes in connection with the mobile node is measured. We use the same notion of neighbor previously explained in the experiment scenario. Figure 4.46 illustrates, for different numbers of objectives, the average number of neighbors measured for all nodes in connection with the mobile node.

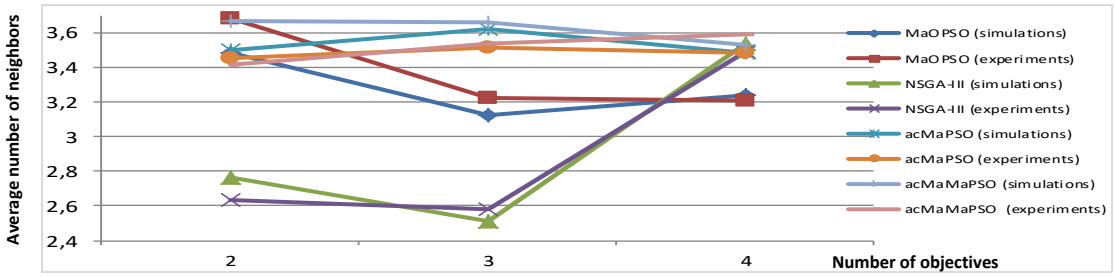


Figure 4.46 Average number of neighbors of nodes in connection with the mobile node

*** Comparing the energy consumption and the network lifetime:** Figure 4.47 shows the energy variations of the network according to the time. Indeed, the average of the energy indicator of all the nodes is measured after adding the nomad nodes.

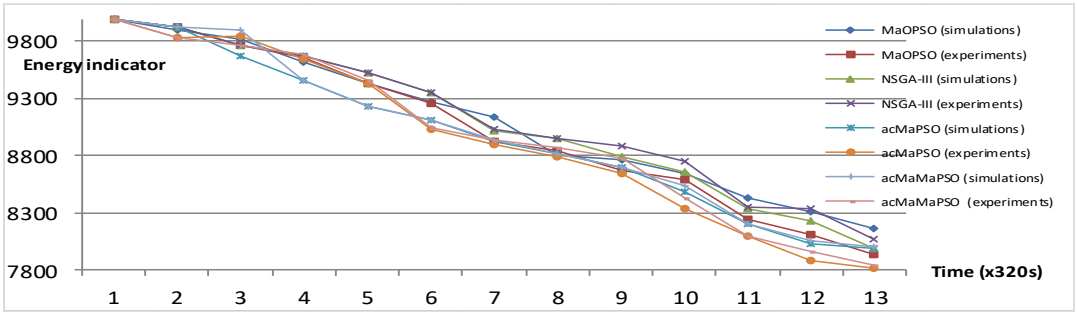


Figure 4.47 Comparing the average energy consumption levels

Table 4.46 shows the lifetime of the network. It illustrates for two numbers of objectives, the time in which the first node is dead.

Table 4.46 Comparing the average lifetime

Nbr of objectives	MaOPSO		NSGA-III		acMaPSO		acMaMaPSO	
	simulations	experiments	simulations	experiments	simulations	experiments	simulations	Experiments
2	3546	3502	3485	3487	3543	3549	3572	3583
5	3478	3469	3528	3546	3540	3553	3568	3571

4.7.2.4 Discussion and interpretations

After evaluating the experiments, different interpretations can be deduced:

- The rates of RSSI and FER are not necessarily inversely proportional: a link between two nodes may have an excellent RSSI and a high FER at the same time.
- The FER values are more relevant during the day than overnight. This can be explained by the activities of persons in the building during experiments such as opening and closing doors, which generates perturbations on the signal.
- In agreement with the findings of different other studies (such as (Figueiredo et al 2016)) showing that the MaOPSO is better than the NSGA-III, our numerical results based on HV and IGD metrics prove that the NSGA-III does not outperforms the MaOPSO. However, experimental results show that our PSO new variants generally outperforms the NSGA-III in the RSSI and FER rates (Thus, it is better in satisfying the coverage, the quality of links and the localization), while the latter is generally more effective than PSO for the number of neighbors of each node (So, it is better in satisfying the utilization of the network). This difference between numerical and experimental results can be explained by the fact that our problem is a real-world one which is different from the theoretical benchmarks used to test the algorithms.
- Since the NSGA-III algorithm is tested by their authors only on instances of theoretical test problems, our study is a proof of the applicability of this algorithm in a real-world context, using real experimentations.
- In contrast with other tested algorithms, the NSGA-III in simulations has higher RSSI values than in experiments.
- In comparison with RSSI, the FER rates are more important in experiments than in simulations.

4.8 Conclusion

In this chapter, we proposed a set of evaluations based on numerical tests, on simulations and on experimentations with different scenarios to prove the effectiveness of the proposed hybridizations on different contexts. To sum up the findings, these evaluations proves that the proposed hybrid algorithms achieve (and surpasses for specific evaluation metrics such as the number of neighbors), the performance of the standard versions of the PSO, MOEA/DD and the NSGA-III algorithms. These findings are summarized in the next chapter (see the section *Findings and results*). Nevertheless, our evaluations can be enhanced by scaling up and testing the deployment proposed by the hybrid algorithms using a larger number of nodes.

Conclusions and future research directions

In this thesis, we addressed and studied the problem of 3D indoor deployment of nodes in collection networks using a multi-objective optimization approach. In fact, when solving real-world problems known by a large number of objectives, the performance of EMOs deteriorates and different outcomes are encountered such as the dependency on the number of objectives, the exponential increase in the number of needed solutions and the size of the population. As a solution to issues, we have studied then modified different hybrid meta-heuristics to improve their performance. We proposed the hybridization of these meta-heuristics both with a dimensionality reduction method, with a method of incorporating user preferences and with the MAS. We tested the positioning solution proposed by these hybrid algorithms with simulations and experiments following different scenarios.

1 Findings and results

The main contributions and corresponding findings of our research work could be summarized as follows:

- An in-depth literature review of methods for optimizing deployment in WSN, particularly the algorithms that take into account 3D indoor deployment. This study covers swarm-based meta-heuristics (Optimization by Particle Swarm, ant colony optimization), genetic algorithms, taboo search, simulated annealing. The study also deals with the single-objective and multi-objective case; the static and dynamic case of deployment.
- A mathematical formulation that aims at modeling the problematic. It is about fixing the objective function to be optimized, the decision variables and the different constraints to take into consideration. Our objective is to minimize the number of used sensor nodes and the consumption of energy. Simultaneously, maximizing the network lifetime, the coverage, the localization and the connectivity.
- We also propose an adaptive mutation and recombination operators that encourages mating with the neighborhood, integrate a concept of multiple scalarizing functions in pareto-based EMOs to overcome the inefficiency of recombination and to increase costs in time and space in the case of many-objective real problems. The obtained results show that for most cases, the aggregation algorithm (MOEA/DD) is more efficient than tested other algorithms (∈-NSGA-II and NSGA-III) to solve the 3D indoor deployment. Moreover, the results prove that the adaptive method of selecting recombination and mutation operators with neighborhood restrictions improves the algorithm performance.
- A first justified hybridization is proposed. It combines two procedures based on user preferences (PI-EMO-PC) and reduction of dimensionality (MVU-PCA) with optimization algorithms belonging to three classes: aggregation-based algorithms (Two_Arch2), reference points-based (MOEA/DD, NSGA-III) and decomposition-based (MOEA/D). To evaluate the proposed new hybrid algorithms, we used the HV metric on our deployment problem; and the IGD metric on theoretical test problems such as DTLZ. Then, to validate the theoretical observations, a set of prototypes were realized. The results show the superior performance of the modified hybrid algorithms over original ones. In addition, there have been other interesting findings such as the superior performance of MOEA/D over NSGA-III in the case of correlated objectives.
- A second hybridization is proposed. It combines an interactive approach of incorporating user preferences (PI-EMO-VF) with the NSGA-III algorithm. Results showed that the performance of the proposed hybrid algorithm exceeds the performance of another recent many-objective algorithm (MOEA/DD) for different evaluation metrics (such as the number of neighbors) although the literature (see (Li et al., 2015) and (Yuan et al., 2016)) states that MOEA/DD gives better results than the standard NSGA-III on theoretical problems.

- A third hybridization is proposed, which combines, for the first time, the ant colony optimization (ACO) algorithm with a recent variant of genetic algorithms (NSGA-III). The advantage of this hybridization scheme is that, unlike conventional hybridizations, the two hybridized algorithms run at the same time and can interact with the same population. From results, it has been deduced that the proposed algorithm (acNSGA-III) outperforms the traditional NSGA-III and the standard ACO.
- Another proposed modification consists in introducing a new concept of bird's accent in the particle swarm optimization algorithm. The new algorithm is named acMaPSO. This new accent concept is based on the idea of separating the swarm (the bird community) into different groups of regional dwellings according to their accents. Thus, to maintain the diversity of the population during the search process of the algorithm, this accent concept evaluates the local particle search capability and classifies particles into different accent groups according to their common way of singing. To escape the local optima, aged particles "die" and are replaced periodically by new randomly generated ones.
- A fourth proposed hybridization consists of distributing the acMaPSO algorithm by hybridizing it with a multi-agent system (MAS). The new algorithm (acMaMaPSO) benefits from the interactivity of particles (becoming agents) and the distribution. The proposed multi-agent architecture is based on three types of agents that have different knowledge and action plans: an environment agent, bird agents (or particle agents) and swarm agents. The acMaPSO and acMaMaPSO algorithms are very competitive compared with the MOPSO algorithm. The obtained results show that acMaPSO and acMaMaPSO exceed for specific evaluation metrics such as the number of neighbors, the performance of the standard PSO algorithm and the NSGA-III algorithm.

To sum up, the performance indicators used to assess our work, the simulations and the real prototyping highlight the good behavior of our proposed approaches on many-objective problems and show the effectiveness of the hybridization.

2 Future research directions

The perspectives of our work are multiple and several open questions remain to be explored. Our perspectives mainly concern two categories: The first category is interested in the development of new hybridization schemes between EMO and other deployment resolution approaches. The second one concerns issues that are related to the network, real experiments and prototyping.

Hybridization perspectives:

- According to its authors (Deb et al. (2010), the PI-EMO-VF procedure contains a large number of parameters to be initialized by the user from the beginning. This makesPI-EMO-VF sensitive to the initial setting. Hence, it is important to propose a parameter-less version of the PI-EMO-VF.
- Although our reduction and preferences platform reduces the number of objectives and the complexity of the problem, it seems to be complex to implement. Hence, to show the contribution of its use, a study of the algorithmic complexity of our hybrid approach can be conducted.
- To evaluate their performances, different recent MOEA and other reduction procedures (rather than the MVU-PCA already used) can be integrated into our platform. In this regard, we propose to incorporate the KnEA algorithm (Zhang et al., 2015) and the FS (Feature Selection) reduction procedure (Mitra et al., 2002).
- The integration of preferences into metaheuristics other than EAs is a promising research direction since different population-based metaheuristics have proven their performance in resolving MOPs. The PSO and the Artificial Immune System (AIS) can be used in this regard.

- In order to help the DM navigate on multidimensional spaces of many-objective problems, researches can be provided regarding the development of better visualization tools.
- A comparative study between all the proposed schemes of hybridization can be implemented.
- As a continuation of this work, we are actually proposing a new optimization algorithm based on the imitation of the behaviors of wild animals in their territories, and its application on our problematic.

Network Application and prototyping Perspectives:

- One of the advantages of our Ophelia testbed is that it is more realistic in terms of constraints and conditions of use than a platform with a large number of uniform nodes such as SmartSantander (Sanchez et al., 2014) , IoTLab (Fleury et al., 2015) or INDRIYA (Doddavenkatappa et al., 2011). These platforms allow us to test our algorithms using a larger number of nodes (up to 1024 nodes). In future studies, in order to compare our Ophelia results with other prototyping platforms and to prove the scalability of our approach, tests on the IoTLab platform are possible since it allows to test the same metrics of our experiences (link quality, RSSI, ... etc)
- Other directions of research can be investigated such as the resolution of dynamic redeployment of nodes with consideration of other objectives. We also aim to intensify the deployed network by adding new nodes to better satisfy the localization constraint that requires four neighbors for each target. We should then redo our experiences to investigate the influence of the network density on the results.
- Another future contribution concerns the implementation of a more realistic energy model with OMNeT ++, based on the BO and SO management values (Farhad et al., 2016) of the CSMA/CA 802.15.4 protocol for example.
- Despite its effectiveness, OpenWiNo lacks libraries that implement standard protocols. Hence the interest of implementing other transmission protocols and technologies in OpenWiNo.

Finally, given the success of multi-objective optimization in solving several real-world problems, often with conflicting goals, the time has come to push its use in different dynamic and complex application domains.

Publications (within this thesis)

International Journals

• **IRJIE'15** MNASRI S., NASRI N., VAN DEN BOSSCHE A., VAL T. **The 3D Deployment Multi-objective Problem in Mobile WSN: Optimizing Coverage and Localization.** International Research Journal of Innovative Engineering. ISSN 2395-0560, Vol. 1 N. issue 5, Mai 2015.

• **AJSE'17** MNASRI S., NASRI N., VAN DEN BOSSCHE A., VAL T. **Resolving the 3D indoor deployment problem using hybrid modified many-objective optimization algorithms.** Arabian Journal for Science and Engineering. ISSN: 2193-567X (Print), (https://link.springer.com/journal/13369). **(Impact Factor= 0.865)**, [**in revision**: submitted on March 31th, 2017 ; ; First demanded revisions received on June 02th, 2018].

• **ISATRANS'18** MNASRI S., NASRI N., VAN DEN BOSSCHE A., VAL T. **A new multi-agent particle swarm algorithm based on birds accents for the 3D indoor deployment problem.** ISA Transactions. ISSN:0019-0578. (https://www.journals.elsevier.com/isa-transactions) **(Impact Factor:3.39, 5Y-IF: 3.4)**, [**in revision**: submitted on February 07th, 2018 ; ; First demanded revisions received on May 04th, 2018].

• **NACOJ'18** MNASRI S., NASRI N., VAN DEN BOSSCHE A., VAL T. **Enhanced hybrid many-objective evolutionary algorithms for real world problems: application to the 3D indoor redeployment problem.** Natural Computing. ISSN 0305-054. (https://link.springer.com/journal/11047) ISSN: 1567-7818 (Print) 1572-9796 (Online). **(Impact Factor= 0.778)**, [**under review**, submitted on February 14th, 2018].

International Conferences

• **PEMWN'14** MNASRI S., NASRI N., VAL T. **An Overview of the deployment paradigms in Wireless Sensor Networks**.International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN 2014), Tunisie – 04-07 Novembre 2014.

• **ISNCC'15** MNASRI S., THALJAOUI A., NASRI N., VAL T. **A genetic algorithm-based approach to optimize the coverage and the localization in the wireless audiosensors networks**. IEEE International Symposium on Networks, Computers and Communications (ISNCC 2015), Hammamet, Tunisie, IEEExplore digital library, 13-15 Mai 2015.

• **IINTEC'17** Mnasri S., Nasri N., VAN DEN BOSSCHE A., Val T. "**A Comparative analysis with validation of NSGA-III and MOEA/D in resolving the 3D indoor redeployment problem in DL-IoT**". IEEE International Conference on Internet of Things, Embedded Systems and Communications (IINTEC'17), Gafsa, Tunisia, October 20-22th, 2017**(Among best six papers)**.

• **AdHocNow'17(a)** Mnasri S., Nasri N., VAN DEN BOSSCHE A., Val T., "**A hybrid ant-genetic algorithm to solve a real deployment problem: a case study with experimental validation**". International Conference on Ad Hoc Networks and Wireless (AdHoc-Now 2017), Messina, Italy,September 20-22 th, 2017, SPRINGER (Eds.), **(Ranked B)**.

• **AdHocNow'17(b)** Mnasri S.,VAN DEN BOSSCHE A. , Nasri N., Val T., " **The 3D redeployment of nodes in Wireless Sensor Networks with real testbed prototyping**". International Conference on Ad Hoc Networks and Wireless (AdHoc-Now 2017), Messina, Italy,September 20-22 th, 2017, SPRINGER (Eds.), **(Ranked B)**.

• **IWCMC'18(a)** MNASRI S., NASRI N., VAN DEN BOSSCHE A., VAL T. **3D indoor redeployment in IoT collection networks: a real prototyping using a hybrid PI-NSGA-III-VF.** The 14th International Wireless Communications and Mobile Computing Conference (http://iwcmc.org/2018/) **(Ranked B)**.

• **IWCMC'18(b)** MNASRI S., NASRI N., VAL T. **The 3D indoor deployment in DL-IoT with experimental validation using a particle swarm algorithm based on the dialects of songs.** The 14th International Wireless Communications and Mobile Computing Conference (http://iwcmc.org/2018/) **(Ranked B)**.

National Conferences

• **JNCT'14** MNASRI S., NASRI N., VAL T. **Contribution au déploiement optimisé des réseaux de capteurs sans fil**. Journées Nationales des Communications Terrestres (JNCT 2014), ISBN 978-3-8417-3468-6, Toulouse-Blagnac, France - 22,23 Mai 2014.

• **JNCT'16** MNASRI S., NASRI N., VAN DEN BOSSCHE A., VAL T. **Simulation d'un réseau sans fil d'intérieur et des algorithmes NSGA-II et NSGA-III modifiés pour la résolution de la problématique de couverture et de localisation 3D**. Journées Nationales des Communications Terrestres (JNCT 2016), Montbéliard – France, 01-02 Septembre 2016.

Bibliography

Ahmad et al. (2007). Ahmad R, Lee YC, Rahimi S, Gupta B. A Multi-Agent Based Approach for Particle Swarm Optimization. International Conference on Integration of Knowledge Intensive Multi-Agent Systems. Waltham, MA, 2007, p. 267-271. DOI: 10.1109/KIMAS.2007.369820

Aitsaadi (2010) N. Ait Saadi. Thesis: Multi-Objective Wireless Sensor NetworkDeployment. PIERRE & MARIE CURIE UNIVERSITY, Ecole Doctorale en Informatique, Télécommunication et Electronique (EDITE of Paris), defense: March, 11th 2010.

Aitsaadi et al. (2008). N. Aitsaadi, N. Achir, K. Boussetta and G. Pujolle, "A Tabu Search Approach for Differentiated Sensor Network Deployment," 2008 5th IEEE Consumer Communications and Networking Conference, Las Vegas, NV, 2008, pp. 163-167. doi: 10.1109/ccnc08.2007.43

Aitsaadi et al., (2011). Artificial potential field approach in WSN deployment: Cost, QoM, connectivity and lifetime constraints. Computer Networks, 55(1): 84-105.

Akyildiz et al. (2007). I.F. Akyildiz, T. Melodia, K.R. Chowdhury, A survey on wireless multimedia sensor networks, Computer Networks 51 (4) (2007) 921–960.

Alia and Al-Ajouri (2017). Alia, O.M.; Al-Ajouri, A.: Maximizing Wireless Sensor Network Coverage With Minimum Cost Using Harmony Search Algorithm. IEEE Sens. J. Vol. 17, no. 3, pp. 882-896 (2017). doi: 10.1109/JSEN.2016.2633409

Ammari M. Habib (2014). The Art of Wireless Sensor Networks Volume 2: Advanced Topics and Applications. DOI 10.1007/978-3-642-40066-7

Arduino (2018). The Arduino platform: Software available at the URL: https://www.arduino.cc/en/main/software (Accessed: January 5th, 2018)

Ateş et al. (2017). E. Ateş, T. E. Kalayci and A. Uğur, "Area-priority-based sensor deployment optimisation with priority estimation using K-means," in IET Communications, vol. 11, no. 7, pp. 1082-1090, 5 11 2017. doi: 10.1049/iet-com.2016.1264

Aval et al. (2012). K. J.Aval and S. Abd Razak, “A Review on the Implementation of Multiobjective Algorithms in Wireless Sensor Network,” World Applied Sciences Journal 19 (6): pp. 772-779, ISSN 1818-4952, 2012; DOI: 10.5829/idosi.wasj.2012.19.06.1398

Bai et al. (2006). X. Bai, S. Kuma, D. Xua, Z. Yun, and T. Lai, “Deploying wireless sensors to achieve both coverage and connectivity,” MobiHoc ’06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, pp. 131–142, 2006.

Banimelhem et al. (2013). O. Banimelhem, M. Mowafi, and W. Aljoby, “Genetic Algorithm Based Node Deployment in Hybrid Wireless Sensor Networks,” Communications and Network, 5, 273-279. Published Online November 2013 (http://www.scirp.org/journal/cn) http://dx.doi.org/10.4236/cn.2013.54034

Baouche et al. (2009). C. Baouche, A. Freitas, and M. Misson, “Radio proximity detection in a WSN to localize mobile entities within a confined area,” Journal of Communications, vol. 4, no. 4, May 2009.

Bellman (1986). R.E. Bellman. « The Bellman continuum ». Editions Robert S. Roth.

Ben Hadj et al. (2006). A. Ben Hadj Mohamed and T. Val, "Interconnexion d'un réseau IP et d'un réseau domotique KNX pour l'aide au maintien à domicile intelligent," Journées Nationales des Communications Terrestres (JNCT2014)- IUT de Toulouse, Université de Toulouse 2,France, May, 22-23th, 2014.

Bechikh et al. (2011). Bechikh S, Ben Said L, Ghédira K (2011) Searching for knee regions of the Pareto front using mobile reference points. Soft Computing – A Fusion of Foundations, Methodologies and Applications, 15(9):1807–1823

Beume et al. (2009). Beume N, Fonseca CM, López-Ibáñez M, Paquete L, Vahrenhold J. On the complexity of computing the hypervolume indicator. IEEE Trans on Evol Comput 13(5):1075–1082. https://doi.org/10.1109/TEVC.2009.2015575

Boufares et al. (2015). N.Boufares, I. Khoufi, P. Minet, L. Azouz Saïdane, Y. Ben Saied: Three dimensional mobile wireless sensor networks redeployment based on virtual forces. IWCMC 2015: 563-568

Bourazza (2006). Variantes d'algorithmes génétiques appliquées aux problèmes d’ordonnancement. Thèse de Doctorat de l'Université du Havre.

Boussaid (2013). I. BOUSSAÏD, “thesis: Perfectionnement de metaheuristiques pour l’otimisation continue,” Université des sciences et de la technologie Houari Boumediene, defense: June, 29th 2013.

Bouzoualegh et al. (2005). Bouzoualegh, T. VAL, E. CAMPO, F. PEYRARD. Modelling and Simulation of Underwater Acoustics Communication based on Stateflow and Simulink Models. Sciences of Electronic, Technologies of Information and Telecommunications (SETIT) Sousse, Tunisia.

Bouzoualegh et al. (2003). Bouzoualegh, T. VAL, F. PEYRARD, E. CAMPO. Study of the Characteristics Needed for Underwater Acoustic Networks. International Conference on CIRCUITS, SYSTEMS, COMMUNICATIONS and COMPUTERS. Corfu, Greece.

Brinis and Saidane (2013). N. Brinis and Leila Azouz Saidane , "Precision Agriculture using WSN," International Workshop on Performance Evaluation and Modeling in Wireless Networks PEMWN 2013. November 11-15, Hammamet, Tunisia.

Britto and Pozo (2012). Britto A, Pozo A. I-MOPSO: A Suitable PSO Algorithm for Many-Objective Optimization. Brazilian Symposium on Neural Networks, Curitiba, p. 166-171. DOI: 10.1109/SBRN.2012.20

Brockhoff et al. (2006). Brockhoff D, Zitzler E. Are All Objectives Necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization. In: Runarsson T.P., Beyer HG., Burke E., Merelo-Guervós J.J., Whitley L.D., Yao X. (eds) Parallel Problem Solving from Nature - PPSN IX. Lecture Notes in Computer Science, vol 4193. Springer, Berlin, pp. 533–542. https://doi.org/10.1007/11844297_54

Brown et al. (2016). Tisha Brown, Zhonghui Wang, Tong Shan, Feng Wang, Jianxia Xue On Wireless Video Sensor Network Deployment for 3D Indoor Space Coverage

Bulusu et al. (2000). N. Bulusu, J. Heidemann, and D. Estrin, “GPS-less low-cost outdoor localization for very small devices,” IEEE Personal Communications, vol. 7, no. 5, pp. 28–34.

Cao et al. (2018). B. Cao, J. Zhao, P. Yang, Z. Ge Lv, X. Liu and G. Min, "3D Multi-Objective Deployment of an Industrial Wireless Sensor Network for Maritime Applications Utilizing a Distributed Parallel Algorithm," in IEEE Transactions on Industrial Informatics. doi: 10.1109/TII.2018.2803758

Carabelea et al. (2004). Carabelea C., Boissier O., Florea A. Autonomy in Multi-agent Systems: A Classification Attempt. In: Nickles M., Rovatsos M., Weiss G. (eds) Agents and Computational Autonomy. AUTONOMY 2003. Lecture Notes in Computer Science, vol 2969. Springer, Berlin, Heidelberg. DOI: https://doi.org/10.1007/978-3-540-25928-2_9

Cardei and Wu (2006). M. Cardei and J. Wu, “Energy-efficient coverage problems in wireless ad-hoc sensor networks,” Florida Atlantic University, Boca Raton, FL 33431, USA, Computer Communications 29, 413–42, 2006

Carlier and Pinson (1989). Carlier J., Pinson E. An algorithm for solving the Job-shop problem. Management Science, N° 35, pp 164-176

Charon et al. (1996). I. Charon, A. Germa, O. Hudry. Méthodes d’optimisation combinatoire. Edition Masson, France

Chen and Jiang (2016). Y. C. Chen and J. R. Jiang, "Particle Swarm Optimization for Charger Deployment in Wireless Rechargeable Sensor Networks," 2016 26th International Telecommunication Networks and Applications Conference (ITNAC), Dunedin, 2016, pp. 231-236. doi: 10.1109/ATNAC.2016.7878814

Cheng et al. (2008). Cheng X, Du D-Z, Wang L, Xu B. Relay sensor placement in wireless sensor networks. Wireless Networks 2008b;14:347–55, doi:10.1007/s11276-006- 0724-8

Clouqueur et al., (2002) Clouqueur, T., Phipatanasuphorn, P.R., Saluja K.K., Sensor Deployment Strategy for Target Detection, Proceedings of 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA ’02)

Coello. (1998). Coello, C. Using the min-max method to solve multiobjective optimization problems with genetic algorithms. In IBERAMIA’98, LNCS. Springer-Verlag.

Coello et al. (2002). Coello CAC, Van Veldhuizen DA, Lamont GB (2002) Evolutionary algorithms for solving multi-objective problems. 1st ed. Springer. https://doi.org/10.1007/978-1-4757-5184-0

Collette and Siarry (2002). Collette.Y and Siarry.P, Optimisation Multiobjectif, Eyrolles

Danping et al. (2013). Danping, H.; Portilla, J.; Riesgo, T.: A 3D multi-objective optimization planning algorithm for wireless sensor networks. In: Proceedings of the 39th Annual Conference of the IEEE Industrial Electronics Society, IECON 2013, p.5428-5433. doi: 10.1109/IECON.2013.6700019

Deb K (2001). Multi-objective optimization using evolutionary algorithms. John Wiley and Sons, Ltd, New York, USA.

Deb et al. (2006). Deb K, Chaudhuri S, Miettinen K. Towards estimating nadir objective vector using evolutionary approaches. 8th Genetic and Evolutionary Computation Conference (GECCO), p 643–650. https://doi.org/10.1145/1143997.1144113

Deb and Jain. (2014). Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference point- based non-dominated sorting approach, part I: Solving problems with box constraints. IEEE Trans Evol Comput 18(4):577-601. https://doi.org/10.1109/TEVC.2013.2281535

Deb et al. (2002). Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T., "A fast and elitist multiobjective genet-ic algorithm: NSGA-II", IEEE Transactions on Evolutionary Computation, 6(2), pp. 182-197. Doi: 10.1109/4235.996017

Deb and Saxena (2006). Deb K, Saxena DK. Searching For Pareto-Optimal Solutions Through Dimensionality Reduction for Certain Large-Dimensional Multi-Objective Optimization Problems. IEEE Congress on Evol Comput, p.3353–3360

Deb et al. (2010) Deb K, Sinha A, Korhonen PJ, Wallenius J. An Interactive Evolutionary Multiobjective Optimization Method Based on Progressively Approximated Value Functions. IEEE Trans on Evol Comput 14(5):723-739, https://doi.org/10.1109/TEVC.2010.2064323

Deb et al. (2005). Deb K, Thiele L, Laumanns M, Zitzler E. Scalable test problems for evolutionary multiobjective optimization. Evolutionary Multiobjective Optimization. In: Abraham A, Jain L, Goldberg R, editors. Advanced Information and Knowledge Processing, London, U.K.: Springer, p. 105–145

Deif and Gadallah (2017). D. S. Deif and Y. Gadallah, "An Ant Colony Optimization Approach for the Deployment of Reliable Wireless Sensor Networks," in IEEE Access, vol. 5, pp. 10744-10756, 2017. doi: 10.1109/ACCESS.2017.2711484

Díaz-Manríquez et al. (2016). Díaz-Manríquez A, Toscano G, Barron-Zambrano JH, Tello-Leal E. R2-Based Multi/Many-Objective Particle Swarm Optimization. Comput Intell and Neuroscience 2016:1898527. DOI:10.1155/2016/1898527

Doddavenkatappa et al. (2011). M. Doddavenkatappa, M.C. Chan and A.L. Ananda, “Indriya: A Low-Cost, 3D Wireless Sensor Network Testbed,” In: Korakis, T. Li, H. Tran-Gia, P. Park, HS., (eds) Testbeds and Research Infrastructure. Development of Networks and Communities, TridentCom, 2011. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 90, Springer, Berlin, Heidelberg, doi:10.1007/978-3-642-29273-6_23

Dorigo and Di Caro (1999). The Ant Colony Optimization Meta-Heuristic, In: D. Corne, M. Dorigo and F. Glover Editors, New Ideas in Optimization, McGraw-Hill.

Dorigo et al. (1996). Dorigo, M., Maniezzo, V. and Colorni, A., "Ant system: optimization by a colony of cooper-ating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)", 26, 1, February 1996, pp. 29-41. Doi: 10.1109/3477.484436

Drechsler et al. (2015). Drechsler N, Sülflow A, Drechsler R. Incorporating user preferences in many-objective optimization using relation e-preferred. Nat Comput 14:469. DOI : 10.1007/s11047-014-9422-0

Eldor (2012). Abbas EL DOR. Thesis: Perfectionnement des algorithmes d’Optimisation par Essaim Particulaire.Applications en segmentation d’images et en électronique, University of Paris-Est, defense: december 5th, 2012

Enayatifar et al. (2014). Enayatifar, R., Yousefi, M., Abdullah, A.H., Darus,A.N.: A Novel Sensor Deployment Approach Using Multi-Objective Imperialist Competitive Algorithm in Wireless Sensor Networks. Arab J Sci Eng. 39(6), 4637-4650 (2014). doi:10.1007/s13369-014-0969-y

Esquirol and Lopez (2001). P. Esquirol, P. Lopez . Concepts et méthodes de base en ordonnancement de la production. Ordonnancement de la production, Edition Hermès, Paris

Farhad et al. (2016). Farhad, A., Farid, S., Zia, Y. and Hussain, F. B. , "A delay mitigation dynamic scheduling algorithm for the IEEE 802.15.4 based WPANs", in Proceedings International Conference on Industrial Informatics and Computer Systems (CIICS), Sharjah, UAE, pp. 1-5. Doi: 10.1109/ICCSII.2016.7462430

Farreny and Ghallab (1987). H. Farreny, M. Ghallab. Eléments de l’intelligence artificielle. Edition Hermès, Paris

Ferentinos et al. (2007). K. P. Ferentinos and T. A. Tsiligiridis, “Adaptive design optimization of wireless sensor networks using genetic algorithms”, Comput. Netw., vol. 51, no. 4, pp. 1031–1051, 2007.

Figueiredo et al. (2016). Figueiredo EMN, Araújo DRB, Filho CJAB, Ludermir TB. Physical topology design of optical networks aided by many-objective optimization algorithms. 5th Brazilian Conference on Intelligent Systems (BRACIS), Recife, p. 409-414. DOI:10.1109/BRACIS.2016.080

Figueiredo et al. (2016). Figueiredo EMN, Ludermir TB, Bastos-Filho CJA. Many Objective Particle Swarm Optimization, Information Sciences p.374:115-134. DOI:10.1016/j.ins.2016.09.026

Fleury et al. (2015). E. Fleury, N. Mitton, T. Noel and C. Adjith, “FIT IoT-LAB: The Largest IoT Open Experimental Testbed,” ERCIM News, ERCIM, 2015, pp.4

Fofana et al. (2016). Nezo Ibrahim Fofana, Adrien van den Bossche, Réjane Dalcé, Thierry Val. “An Original Correction Method for Indoor Ultra-Wide Band Ranging-based Localisation System” Springer 15th International Conference on Ad Hoc Networks and Wireless – AdHoc-Now, Lille, France.

Fonseca et al.(2006). Fonseca CM, Paquete L, López-Ibáñez M. An improved dimension - sweep algorithm for the hypervolume indicator. Congress on Evolutionary Computation, p. 1157–1163. IEEE Press, Piscataway. https://doi.org/10.1109/CEC.2006.1688440

Frye et al. (2006). Frye, L., Cheng, L., Du, S. & Bigrigg, M. W. (2006). Topology maintenance of wireless sensor networks in node failure-prone environments. Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control, (pp. 886-891). Piscataway, NJ, USA: IEEE.

Fujito et Kurahashi, (2006). Fujito, T. & Kurahashi, H. (2006). A Better-Than-Greedy algorithm for k-set multicover. 3rd International Workshop on Approximation and Online Algorithms, (3879 LNCS, pp. 176–189). Berlin, Germany: Springer-Verlag

Gandibleux et al. (1996). Gandibleux, X., Mezdaoui, N., and Freville, A. A tabu search procedure to solve multiobjective combinatorial optimization problems. In Caballero, R., Ruiz, F., and Steuer, R., editors, Second Int. Conf. on Multi-Objective Programming and Goal Programming MOPGP’96, pages 291–300, Torremolinos, Spain. Springer-Verlag.

Gong et al. (2013) Gong D, Wang G, Sun X. Set-based genetic algorithms for solving many-objective optimization problems. 13th UK Workshop on Computational Intelligence (UKCI), Guildford, p. 96-103. https://doi.org/10.1109/UKCI.2013.6651293Glover (1989). Glover, F. “Tabu Search – Part I”, ORSA Journal on Computing 1, 190-206

Guang et al. (2009). Guang Tan, Stephen A. Jarvis, Member, and Anne-Marie Kermarrec “Connectivity-Guaranteed and Obstacle-Adaptive Deployment Schemes for Mobile Sensor Networks” IEEE transactions on mobile computing, vol. 8, no. 6, June 2009

Guvensan et Yavuz (2011). M. Amac Guvensan, A. Gokhan Yavuz, On coverage issues in directional sensor networks: A survey, Ad Hoc Networks, Volume 9, Issue 7, 2011, Pages 1238-1255, https://doi.org/10.1016/j.adhoc.2011.02.003

Hall and Moberg (2003). Hale, T. S. & Moberg, C. R. (2003). Location science research: a review. Annals of Operations Research, (vol. 123, pp. 21-35). Netherlands: Kluwer Academic Publishers.

Hall (1988). P. Hall, Introduction to the Theory of Coverage Processes. John Wiley and Sons, 1988.

Han et al. (2013). G. Han, C. Zhang, L. Shu, N. Sun, and Q. Li, “A Survey on Deployment Algorithms in Underwater Acoustic Sensor Networks,” Hindawi Publishing Corporation, International Journal of Distributed Sensor Networks; Volume 2013, Article ID 314049, http://dx.doi.org/10.1155/2013/314049

He et al. (2003). T.He, C.Huang, B.M. Blum, J. A. Stankovic, and T. Abdelzaher, “Rangefree localization schemes for large scale sensor networks,” in Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom ’03), pp.

Hertz et al. (1994) A. Hertz, B. Jaumard, M.P. De Aragao, Local Optima Topology for the k-coloring Problem. Discrete Applied Mathematics 49: 257-280

Holland (1975). J.H. Holland. Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan; re-issued by MIT Press (1992)

Hu et al. (2017). Hu W, Yen GG, Luo G. Many-objective particle swarm optimization using two-stage strategy and parallel cell coordinate system. IEEE Trans on Cybern;47(6):1446-1459. DOI: 10.1109/TCYB.2016.2548239

Huang and Chne (2013). Huang, P. and Chen, J., "Improved CCN routing based on the combination of genetic algorithm and ant colony optimization", in Proceedings 3rd International Conference on Computer Science and Network Technology, Dalian, China, October 12-13 2013, pp. 846-849. Doi: 10.1109/ICCSNT.2013.6967238

Hwang and Masud (1979). Hwang C. and Masud A. Multiple objective decision making -methods and applications. In Lectures Notes in Economics and Mathematical Systems, volume 164. Springer-Verlag, Berlin

Ibrahim et al. (2016). Ibrahim, A., Rahnamayan, S., Martin, M. V. and Deb, K., "EliteNSGA-III: An improved evolutionary many-objective optimization algorithm", in Proceedings IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, July 24-29 2016, pp. 973-982. Doi: 10.1109/CEC.2016.7743895

Idoudi (2017). Idoudi Hanene, Contributions à l'amélioration des communications dans les réseaux sans fil multi-sauts, habilitation to supervise researches (HDR) from the university of Toulouse, defended on decembre 1st, 2017

Idoudi et al. (2012). Hanen Idoudi, Chiraz Houaidia, Leila Azouz Saidane, Pascale Minet. Robots-Assisted Redeployment in Wireless Sensor Networks. Journal of Networking Technology, Dline publisher, Vol. 3, No. 1, pp 1-12. http://www.dline.info/jnt/v3n1.php

IoTLab. (2017). IoTLab platform, URL: https://www.iot-lab.info. Accessed April 08th, 2017

Ishibuchi et al. (2015). Ishibuchi, H.; Akedo, N.; Nojima, Y.: Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems. IEEE Trans. on Evol. Comput. 19(2), pp 264-283

Ishibuchi et al (2009). H. Ishibuchi, Y. Sakane, N. Tsukamoto and Y. Nojima, "Adaptation of Scalarizing Functions in MOEA/D: An Adaptive Scalarizing Function-Based Multiobjective Evolutionary Algorithm," 5th International Conference on Evolutionary Multi-Criterion Optimization (EMO2009), LNCS 5467, Nantes, France, April 2009, pp. 438–452, doi:10.1007/978-3-642-01020-0_35

Ishibuchi et al. (2010). Ishibuchi, H.; Sakane, Y.; Tsukamoto, N.; Nojima, Y.: Simultaneous Use of Different Scalarizing Functions in MOEA/D. In: Proceedings of the 12th annual conference on Genetic and Evolutionary Computation GECCO, pp. 519–526

Ishibuchi et al. (2008). Ishibuchi H, Tsukamoto N, Nojima Y. Evolutionary Many-Objective Optimization: A Short Review. IEEE Congress on Evol Comput Hong Kong, p2424–2431. https://doi.org/10.1109/CEC.2008.4631121

Jaimes et al. (2011). Jaimes AL, Coello CAC, Aguirre HE, Tanaka K (2011) Adaptive objective space partitioning using conflict information for manyobjective optimization. Evolutionary Multi-Criterion Opt., Springer, p. 151–165. https://doi.org/10.1007/978-3-642-19893-9_11

Jaimes et al. (2008). Jaimes AL, Coello CAC, Chakraborty D. Objective Reduction Using a Feature Selection Technique. Genetic and Evol Comput Conference (GECCO), p. 673–680. https://doi.org/10.1145/1389095.1389228

Jaimes et al (2011). A. L. Jaimes, A. A. Montaño and C. A. C. Coello, "Preference incorporation to solve many-objective airfoil design problems," IEEE Congress of Evolutionary Computation (CEC2011), New Orleans, LA, June 2011, pp. 1605-1612, doi: 10.1109/CEC.2011.5949807

Jason et al. (2004). Jason T, Shanchieh Y, Raghuveer R, Ferat S. Optimal topologies for wireless sensor networks. International Society for Optical Engineering 2004. p192-203. DOI:10.1117/12.578518

Jiang et al. (2014). Jiang S, Ong YS, Zhang J, Feng L. Consistencies and Contradictions of Performance Metrics in Multiobjective Optimization, IEEE Trans on Cybern;44(12):2391-2404, http://dx.doi.org/ 10.1109/TCYB.2014.2307319

Jiang et al. (2016). Peng Jiang, Shuai Liu, Jun Liu, Feng Wu and Le Zhang, A Depth-Adjustment Deployment Algorithm Based on Two-Dimensional Convex Hull and Spanning Tree for Underwater Wireless Sensor Networks, Sensors 2016, 16(7), 1087; doi:10.3390/s16071087

Jiang et al. (2016). Jiang, J. A.; Wan, J. J.; Zheng, X. Y.; Chen, C. P.; Lee, C. H.; Su, L. K.; Huang, W. C.: A Novel Weather Information-Based Optimization Algorithm for Thermal Sensor Placement in Smart Grid. IEEE Trans. Smart Grid , vol.PP, no.99, pp.1-11 (2016). DOI: 10.1109/TSG.2016.2571220

Jin et al. (2012). M. Jin, G. Rong, H. Wu, L. Shuai, X. Guo, Optimal surface deployment problem in wireless sensor networks. in Proceedings of IEEE INFOCOM 2012. pp. 2345–2353 (2012). DOI: 10.1109/INFCOM.2012.6195622

Kadri et al. (2014). H. Kadri and B. Zouari, "A high-level Petri nets approach for multi-objective optimization in pipeline networks," 2014 4th International Conference On Simulation And Modeling Methodologies, Technologies And Applications (SIMULTECH), Vienna, Austria, 2014, pp. 211-218. doi: 10.5220/0005092602110218

Kang and Chen (2009). Kang, C-W.; Chen, J-H.: An evolutionary approach for multiobjective 3D differentiated sensor network deployment. In: Proceeding of the International Conference on Computational Science and Engineering, CSE'09, Vancouver, BC, pp. 187-193. doi: 10.1109/CSE.2009.329

Kennedy and Eberhart. (1995). J. Kennedy and R. C. Eberhart. "Particle Swarm Optimization". In : Proceedings of the IEEE International Conference on Neural Networks IV, pp. 1942–1948, Perth, Australia

Khalfallah et al. (2017). Zakia Khalfallah, Ilhem Fajjari∗, Nadjib Aitsaadi‡, Paul Rubin† and Guy Pujolle, A Novel 3D Underwater WSN Deployment Strategy for Full-Coverage and Connectivity in Rivers, IEEE ICC 2016 Ad-hoc and Sensor Networking Symposium

Khemiri et al. (2017). R. Khemiri, K. Elbedoui-Maktouf, B. Grabot and B. Zouari, "Integrating fuzzy TOPSIS and goal programming for multiple objective integrated procurement-production planning," 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limassol, 2017, pp. 1-8.doi: 10.1109/ETFA.2017.8247644

Kirkpatrick et al. (1983). S.Kirkpatrick, C.Gelatt, Jr and M.Vecchi, optimization by simulated annealing, Science, vol 220, No 4508, May 1983, pp, 671-680

Ko and Gagnon (2015). Ko, AHR.; Gagnon, F.: Process of 3D wireless decentralized sensor deployment using parsing crossover scheme. Appl Comput Inform, vol 11, issue 2, pp. 89–101 (2015). doi: 10.1016/j.aci.2014.11.001

Kollat et al. (2005). Kollat JB, Reed PM. The value of online adaptive search: A performance comparison of NSGAII, ε -NSGAII and ε-MOEA. In: Evolutionary Multi-Criterion Optimization. Berlin, Germany: Springer- Verlag; pp 386–398

Kolomvatsos and Hadjieftymiades (2014). Kolomvatsos K, Hadjieftymiades S. On the use of particle swarm optimization and kernel density estimator in concurrent negotiations. Information Sciences 2014;262:99–116. DOI: 10.1016/j.ins.2013.10.025

Konstantinidis et al. (2009). A. Konstantinidis, K. Yang, Q. Zhang, and D. Zeinalipour-Yazti, "A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks," Computer Networks 54(6): 960-976, 2010

Korejo (2011). Korejo, I. A.: Adaptive Mutation Operators for Evolutionary Algorithms. Ph.D. diss., Department of Computer Science, University of Leicester

Kumar et al. (2011). Kumar R, Sharma D, Sadu A. A hybrid multi-agent based particle swarm optimization algorithm for economic power dispatch. Int J of Elec Power & Energy Sys 2011;33(1):115-123. DOI:10.1016/j.ijepes.2010.06.021

Kursawe (1991). Kursawe, F. A variant of evolution strategies for vector optimization. In Schwefel, H. and Manner, R., editors, Parallel Problem Solving from Nature, volume 496 of Lecture Notes in Computer Science, pages 193–197, Berlin. Springer-Verlag.

Lan et al. (2010) Lan T, Erdogmus D, Black L, Van Santen J. A comparison of different dimensionality reduction and feature selection methods for single trial ERP detection. Annual International Conference of the IEEE Engineering in Medicine and Biology, Buenos Aires, p. 6329-6332. https://doi.org/doi: 10.1109/IEMBS.2010.5627642

Laumanns et al. (2002). Laumanns M, Thiele L, Deb K, Zitzler E. Combining convergence and diversity in evolutionary multiobjective optimization. Evol Comput 10(3):263-282. https://doi.org/10.1162/106365602760234108

Li et al. (2015). Li K, Deb K, Zhang Q, Kwong S. An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition. IEEE Trans Evol Comput 19(5):694-716. https://doi.org/10.1109/TEVC.2014.2373386

Li et al. (2015). Y. Li, W. Gao, C. Wu and Y. Wang, "Deployment of Sensors in WSN: An Efficient Approach Based on Dynamic Programming," in Chinese Journal of Electronics, vol. 24, no. 1, pp. 33-36, 01 2015. doi: 10.1049/cje.2015.01.006

Lin (2015) P. C. Lin, "Optimal smart gateway deployment for the Internet of Things in smart home environments," 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE), Osaka, 2015, pp. 273-274. doi: 10.1109/GCCE.2015.7398715

Liu and Ouyang (2018). ZHIMIN LIU and ZHANGDONG OUYANG, k-Coverage Estimation Problem in Heterogeneous Camera Sensor Networks With Boundary Deployment, *DOI 10.1109/ACCESS.2017.2785393*

Matsuo et al. (2013). Matsuo, S.; Sun, W.; Shibata, N.; Kitani, T.; Ito, M.: BalloonNet: A deploying method for a three-dimensional wireless network surrounding a building. In: Proceedings of the Eighth International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), pp. 120-127 (2013). doi: 10.1109/BWCCA.2013.28

Mei et al. (2004). Y. Mei, Y.-H. Lu, Y. Hu and C. Lee, "Energy-efficient motion planning for mobile robots," Proceedings of IEEE International Conference on Robotics and Automation (ICRA), vol. 5, pp. 4344-4349, New Orleans, LA, April 26- May 1, 2004

Mekni and Haddad (2010). M. Mekni and H. Haddad, "A Knowledge-Based Multi-agent Geo-simulation Framework: Application to Intelligent Sensor Web Deployment," 2010 Fourth International Conference on Sensor Technologies and Applications, Venice, 2010, pp. 329-335. doi: 10.1109/SENSORCOMM.2010.111

Mnasri et al. (2014a). Mnasri S., Nasri N., Val T. An Overview of the deployment paradigms in the Wireless Sensor Networks.International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN 2014), Tunisia – 04-07th November 2014

Mnasri et al. (2014b). Mnasri S., Nasri N., Val T. Contribution au déploiement optimisé des réseaux de capteurs sans fil. Journées Nationales des Communications Terrestres (JNCT 2014), ISBN 978-3-8417-3468-6, Toulouse-Blagnac - 22,23th May 2014

Mnasri et al. (2017a) Mnasri S., Nasri N., VAN DEN BOSSCHE A., Val T., "A hybrid ant-genetic algorithm to solve a real deployment problem: a case study with experimental validation". International Conference on Ad Hoc Networks and Wireless (AdHoc-Now 2017), Messina, Italy,September 20-22 th, 2017, SPRINGER (Eds.), Springer International Publishing

Mnasri et al. (2017b) Mnasri S.,VAN DEN BOSSCHE A. , Nasri N., Val T., " The 3D redeployment of nodes in Wireless Sensor Networks with real testbed prototyping". International Conference on Ad Hoc Networks and Wireless (AdHoc-Now 2017), Messina, Italy,September 20-22 th, 2017, SPRINGER (Eds.), Springer International Publishing

Mnasri et al. (2017c). Mnasri S, Nasri N, Van Den Bossche A, VAL T. A Comparative analysis with validation of NSGA-III and MOEA/D in resolving the 3D indoor redeployment problem in DL-IoT, IEEE International Conference on Internet of Things, Embedded Systems and Communications (IINTEC 2017), Gafsa, Tunisia. DOI: 10.1109/IINTEC.2017.8325906

Mnasri et al. (2018a) Mnasri S., Nasri N., VAN DEN BOSSCHE A., Val T. The 3D indoor redeployment process in DL-IoT: a real testbed prototyping using an hybrid PI-NSGAIII-VF optimization strategy. The 14th International Wireless Communications and Mobile Computing Conference (in press)

Mnasri et al. (2018b) Mnasri S., Nasri N., VAN DEN BOSSCHE A., Val T. The deployment in 3D indoor DL-IoT with experimental validation using a multi-agent particle swarm optimization algorithm based on birds accents. The 14th International Wireless Communications and Mobile Computing Conference (in press)

Nakrani and Tovey. (2004). Sunil Nakrani Craig Tovey. On Honey Bees and Dynamic Server Allocation in Internet Hosting Centers, Adaptive Behavior , Vol 12, Issue 3-4, pp. 223 - 240, https://doi.org/10.1177/105971230401200308

Nebro et al. (2013). Nebro, A.J.; Durillo, J.J.; Machin, M.; Coello, C.A.C.; Dorronsoro, B.: A Study of the Combination of Variation Operators in the NSGAII Algorithm. C. Bielza et al. (Eds.): CAEPIA2013, LNAI 8109, pp. 269–278. doi:10.1007/978-3-642-40643-0_28

Oesel et al. (2017). A. Oesel, A.C. Fries, L. Miller, H.L. Gibbs, J.A. Soha and D.A. Nelson, “High levels of gene flow among song dialect populations of the Puget Sound white-crowned sparrow,” Ethology, 2017, vol. 123, pp. 581–592, doi: https://doi.org/10.1111/eth.12632

Omnetpp (2018). Omnetpp software. URL: https://omnetpp.org/omnetpp. Accessed January 09th, 2018

O’Rourke (1987). O'Rourke J. Art Gallery Theorems and Algorithms. Oxford University Press, 1987. ISBN 0-19-503965-3

Osman and Laporte (1996). I.H. Osman, G. Laporte, Metaheuristics: A bibliography, Annals of Operations research, 63: 511-623

Papadimitriou and Steiglitz (1982). Papadimitriou, C. H. and Steiglitz, K. Combinatorial Optimization: Algorithms and Complexity. Prentice-Hall

Pei et al. (2008). Z. Pei, Z. Deng, B. Yang and X. Cheng, "Application-oriented wireless sensor network communication protocols and hardware platforms: A survey," IEEE International Conference on Industrial Technology, pp. 1-6, Chengdu, China, April 21-24, 2008

Peng et al. (2008). Peng C, Anbo M, Chunhua Z. Particle swarm optimization in multi-agent system for the intelligent generation of test papers. IEEE Congress on Evolutionary Computation. Hong Kong, 2008, p. 2158-2162. DOI: 10.1109/CEC.2008.4631085

Puchinger and Raidl. (2005). J. Puchinger et G. R. Raidl. Combining metaheuristics and exact algorithms in combinatorial optimization: a survey and classification. Institute of Computer Graphics and Algorithms Vienna University of Technology, Vienna, Austria

Purshouse et al. (2007). Purshouse RC, Fleming PJ. On the evolutionary optimization of many conflicting objectives. IEEE Trans Evol Comput 11(6):770-784. https://doi.org/10.1109/TEVC.2007.910138

Qi et al. (2015). J. Qi, R. Vazquez and M. Krstic, "Multi-Agent Deployment in 3-D via PDE Control," in IEEE Transactions on Automatic Control, vol. 60, no. 4, pp. 891-906, April 2015. doi: 10.1109/TAC.2014.2361197

Qu (2013). Qu Y. thesis: Wireless Sensor Network Deployment,” Florida International University, Miami, Florida, USA

Qu et al. (2012). Qu, B.Y.; Suganthan, P. N.; Liang, J.J.: Differential Evolution with Neighborhood Mutation for Multimodal Optimization. IEEE Trans. Evol. Comput, Vol. 16, No. 5, pp. 601-14

Rachmawati et al. (2006). Rachmawati L, Srinivasan D. Preference incorporation in multi-objective evolutionary algorithms: A survey. IEEE Congress on Evolut Comput (CEC) Vancouver, BC, p962–968. http://dx.doi.org/10.1109/CEC.2006.1688414

Richardson et al. (1989). Richardson, J., Palmer, M., Liepins, G., and Hilliard, M. Some guidelines for genetic algorithms with penalty functions. In Third Int. Conf. on Genetic Algorithms ICGA’3, pages 191–197.

Riquelme at al. (2015). Riquelme N, Von Lucken C, Baran B. Performance metrics in multi-objective optimization. XLI Latin American Computing Conference (CLEI), p XXX, http://dx.doi.org/ 10.1109/CLEI.2015.7360024

Ritzel et al (1994) Ritzel, B., Eheart, J., and Ranjithan, S. Using genetic algorithms to solve a multiple objective groundwater pollution problem. Water Resources Research, 30(5):1589–1603

Romer et Mattern (2004) K. Romer and F. Mattern, “The design space of wireless sensor networks”, IEEE Wirel. Commun., vol. 11, no. 6, pp. 54–61, 2004

RSSI. (2016). Converting the input power to RSSI. URL: https://www.sparkfun.com/datasheets/Wireless/General/RFM22B.pdf. page 57. Accessed February 14th, 2016

Sanchez et al. (2014). L. Sanchez, L. Muñoz, J. A. Galache, S. Pablo, J. R. Santana, V. Gutierrez, et al., “SmartSantander: IoT experimentation over a smart city testbed,” Computer Networks, vol. 61, 2014, pp. 217-238, doi:10.1016/j.bjp.2013.12.020

Sato et al. (2014). Sato., H.: Inverted PBI in MOEA/D and its impact on the search performance on multi and many-objective optimization. In: Proceedings of the 16th Annual Conference on Genetic and Evolutionary Computation (GECCO), pp. 645-652

Saunders and Arag´on-Zavala. (2007). Saunders SR, Arag´on-Zavala A. Antennas and propagation for wireless communication systems. 2nd edition. Wiley

Saxena et al. (2013). Saxena DK, Duro JA, Tiwari A, Deb K, Zhang Q. Objective Reduction in Many-Objective Optimization: Linear and Nonlinear Algorithms. IEEE Trans on Evol Comput 17(1):77-99. https://doi.org/10.1109/TEVC.2012.2185847

Schaffer (1985). Multiple objective optimization with vector evaluated genetic algorithms. In Grefenstette, J., editor, ICGA Int. Conf. on Genetic Algorithms, pages 93–100. Lawrence Erlbaum.

Seada et al. (2016). Seada, H.; Deb, K.: A Unified Evolutionary Optimization Procedure for Single, Multiple, and Many Objectives. IEEE Trans Evol Comput, vol.20, no.03, p.358-369 (2016). doi: 10.1109/TEVC.2015.2459718

Sen et al. (1988). Sen, T., Raiszadeh, M. and Dileepan, P. A branch and bound approach to the bicriterion scheduling problem involving total flowtime and range of lateness. Management Science, 34(2) :254-260

Serafini (1992). Simulated annealing for multiple objective optimization problems. In Tenth Int. Conf. on Multiple Criteria Decision Making, pages 87–96, Taipei

Shen (2016). Shen, H., "A study of welding robot path planning application based on Genetic Ant Colony Hybrid Algorithm", in Proceedings IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, China, October 3-5 2016, pp. 1743-1746. Doi: 10.1109/IMCEC.2016.7867517

Shen et al. (2013). Y. Shen, D. T. Nguyen, and M. T. Thai, “Adaptive Approximation Algorithms for Hole Healing in Hybrid Wireless Sensor Networks,” in Proceedings of the 32nd Int Conference on Computer Communications (INFOCOM), 2013

Shi and Eberhart (1999). Shi Y, Eberhart R. Empirical study of particle swarm optimization. Congress on Evolutionary Computation. Washington, DC, p. 1950 vol. 3. DOI: 10.1109/CEC.1999.785511.

Sim et al. (2003). Sim, K. M. and Sun, W. H., "Ant colony optimization for routing and load-balancing: survey and new directions", IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 33, 5, November 17 2003, pp. 560-572. Doi: 10.1109/TSMCA.2003.817391

Singh et al. (2011). Singh HK, Isaacs A, Ray T. A Pareto Corner Search Evolutionary Algorithm and Dimensionality Reduction in Many Objective Optimization Problems. IEEE Trans on Evol Comput 15(4):539–556. https://doi.org/10.1109/TEVC.2010.2093579

Sinha et al. (2014). Sinha A, Korhonen P, Wallenius J, Deb K. An Improved Progressively Interactive Evolutionary Multi -objective Optimization Algorithm with a Fixed Budget of Decision Maker Calls. European J of Oper Research 233(3):674–688, https://doi.org/10.1016/j.ejor.2013.08.046

Sinha at al. (2013). Sinha A, Saxena DK, Deb K, Tiwari A. Using objective reduction and interactive procedure to handle many-objective optimization problems. Appl Soft Comput 13(1):415-427. https://doi.org/10.1016/j.asoc.2012.08.030

Stewart and White (1991). Stewart, B. and White, C. Multiobjective a*. Journal of the ACM, 38(4) :775–814

Sweidan and Havens (2016). Sweidan, H.I.; Havens, T. C.: Coverage optimization in a terrain-aware wireless sensor network. In: Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, pp. 3687- 3694 (2016). doi: 10.1109/CEC.2016.7744256

Talbi (1999). E.G. Talbi. Métaheuristiques pour l’optimisation combinatoire multiobjectif. Rapport C.N.E.T (France Télécom) Paris

Talbi. (2000). E. G. Talbi. Une taxinomie des métaheuristiques hybrides. ROADEF 2000

Tan et al. (2010). G. Tan, H. Jiang, S. Zhang, and A.-M. Kermarrec, “Connectivity- based and anchor-free localization in large-scale 2D/3D sensor networks,” in Proceedings of the 11th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc ’10), pp. 191–200.

Tian et al. (2017). Tian Y, Cheng R, Zhang X, Jin Y. PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization. IEEE Comput Intell Mag 12(4):73-87. https://doi.org/10.1109/MCI.2017.2742868

Tian et Georganas (2004). Tian D, Georganas N. Connectivity maintenance and coverage preservation in wireless sensor networks. In: Canadian conference on electrical and computer engineering, 2004, vol.2; May 2004. p.1097–100

Tan et al. (2013). Tan, Y.; Jiao, Y.; Li, H.; Wang. X.: MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives. Comput. & Oper. Res., 40(6), pp.1648-1660

Topcuoglu et al. (2011). Topcuoglu, H.R.; Ermis, M.; Sifyan, M.: Positioning and utilizing sensors on a 3-D terrain Part II-Solving with a hybrid evolutionary algorithm. IEEE Trans. Human–Mach. Syst., vol.41, no.4, pp.470-480 (2011). doi:10.1109/TSMCC.2010.2055851

Udgata et al (2009). S. K. Udgata, S. L. Sabat and S. Mini, "Sensor deployment in irregular terrain using Artificial Bee Colony algorithm," 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, 2009, pp. 1309-1314. doi: 10.1109/NABIC.2009.5393734

Unaldi et al. (2012). Unaldi, N.; Temel, S.; Asari, VK.: Method for optimal sensor deployment on 3D terrains utilizing a steady state genetic algorithm with a guided walk mutation operator based on the wavelet transform. Sensors, vol. 12, no.4, pp. 5116-5133 doi: 10.3390/s120405116

Val (1994). Thierry VAL. Thesis: Etude d'un reseau local hybride d'interieur permettant l'interconnexion de stations fixes et de stations mobiles, Clermont-Ferrand University, France.

Van den Bossche et al. (2016). Adrien van den Bossche, Réjane Dalcé, Nezo Ibrahim Fofana et Thierry Val “DecaDuino: An Open Framework for Wireless Time-of-Flight Ranging Systems” Wireless Days, Toulouse.

Van den Bossche et al. (2016). Van den Bossche, A., Dalce, R.and Val, T. OpenWiNo: An Open Hardware and Software Framework for Fast-Prototyping in the IoT, in Proceedings 23rd International Conference on Telecommunications (ICT), Thessaloniki, Greece, May 16-18 2016, pp. 1-6. Doi: 10.1109/ICT.2016.7500490

Van den Bossche et al. (2017). Van den Bossche, A.; Dalce, R.; Val., T.: Plateforme de prototypage rapide d'objets connectés avec la famille WiNo -Enabling Fast prototyping of Connected Things using the WiNo family. ISTE OpenScience - Internet des objets, Vol. 1, N.1

Veldhuizen et al. (1997) Veldhuizen, D. V., Sandlin, B., Marmelstein, R., Lamont, G., and Terzuoli, A. Finding improved wire-antenna geometries with genetic algorithms. In Chawdhry, P., Roy, R., and Pant, P., editors, Soft Computing in Engineering Design and Manufacturing, pages 231–240, London. Springer Verlag

Vrugt and Robinson, (2007). Vrugt, J.A.; Robinson, B.A.: Improved evolutionary optimization from genetically adaptive multimethod search. In: Proceedings of the National Academy of Sciences of the United States of America 104 (3), pp. 708– 711. doi:10.1073/pnas.0610471104

Wang et al. (2007) G. G. Wang, G. Cao, P. Berman, and T. F. L. Porta, “Bidding protocols for deploying mobile sensors,” IEEE Transactions on Mobile Computing, vol. 6, no. 5, pp. 563–576, 2007

Wang et al. (2005). G. L. Wang, G. H. Cao, T. LaPorta, and W. S. Zhang, “Sensor relocation in mobile sensor networks,” in *Proc. 24rd Annu. Joint Conf. IEEEComput.Commun. Societies (INFOCOM)*, Miami, FL, Mar. 2005, pp. 2302–2312

Wang et al. (2008). Wang, Y., Hu, C. & Tseng, Y. (2008). Efficient placement and dispatch of sensors in a wireless sensor network. IEEE Transactions on Mobile Computing, 7(2), pp. 262-274

Wang et al. (2015). Wang H, Jiao L, Yao X. Two_Arch2: An Improved Two-Archive Algorithm for Many-Objective Optimization. IEEE Trans on Evol Comput 19(4):524-541. https://doi.org/10.1109/TEVC.2014.2350987

Wang et al. (2003). Wang X, Xing G, Zhang Y, Lu C, Pless R, Gill C. Integrated coverage and connectivity configuration in wireless sensor networks. In: Proceedings of the 1st international conference on embedded networked sensor systems, ser. SenSys’03., New York,NY,USA:ACM;2003.p.28–39

Wickramasinghe et al. (2010). Wickramasinghe UK, Carrese R, Li X. Designing airfoils using a reference point based evolutionary many-objective particle swarm optimization algorithm. IEEE Congress on Evol Comput, Barcelona, p. 1-8. http://dx.doi.org/doi: 10.1109/CEC.2010.5586221

Widmer et al. (2001). M. Widmer, A. Hertz, D. Costa. “Les métaheuritiques” in «ordonnancement de la production » Ed. Pierre lopez and François Roubellat, Hermes science europe Ltd.

Wienke et al. (1992) Wienke, P., Lucasius, C., and Kateman, G. Multicriteria target optimization of analytical procedures using a genetic algorithm. Analytical Chimica Acta, 265(2):211–225

White D. (1982). The set of efficient solutions for multiple-objectives shortest path problems. Computers and Operations Research, 9 :101–107

Wu and Wang (2017). Chase Q. Wu and Li Wang On Efficient Deployment ofWireless Sensors for Coverage and Connectivity in Constrained 3D Space

Wu el al. (2015). Wu C, Li H, Wu L, Wu Z. A Multi-Agent Particle Swarm Optimization for Power System Economic Load Dispatch. J of Comput and Comm 2015;3:83-89. DOI:10.4236/jcc.2015.39009

Wu et al. (2007). Wu Q., Rao, N. S. V., Du, X., Iyengar, S. S. & Vaishnavi, V. K. On efficient deployment of sensors on planar grid. *Computer Communications, 30*(14-15), pp. 2721-2734

Xie and Liu. (2017). Jing Xie, Chen-Ching Liu. Multi-agent systems and their applications, Journal of International Council on Electrical Engineering, 7:1, 188-197, DOI: 10.1080/22348972.2017.1348890

Xu et al. (2014). Xu , H.; Lai, Z.; Liang, H.: A novel mathematical morphology based antenna deployment scheme for indoor wireless coverage. In: Proceedings of the IEEE 80th Vehicular Technology Conference (VTC Fall), pp.1-5, (2014). doi: 10.1109/VTCFall.2014.6965828

Xu et al. (2010) J. Xu, S. Li, and X. Bai, "Mobile Sensor Deployment Optimization for k-Coverage in Wireless Sensor Networks with a Limited Mobility Model," IETE Technical Review, vol. 27, no. 2, pp. 124–137, 2010

Younis et Akkaya (2008). Mohamed Younis, Kemal Akkaya, Strategies and techniques for node placement in wireless sensor networks: A survey. Ad Hoc Networks 6 (2008) 621–655

Yu el al. (2013). X. Yu, N. Liu, W. Huang, X. Qian, and T. Zhang, "A Node Deployment Algorithm Based on Van Der Waals Force in Wireless Sensor Networks," Hindawi Publishing Corporation; International Journal of Distributed Sensor Networks; Volume 2013, Article ID 505710; http://dx.doi.org/10.1155/2013

Yuan et al. (2016). Y. Yuan, H. Xu, B. Wang, B. Zhang and X. Yao, "Balancing Convergence and Diversity in Decomposition-Based Many-Objective Optimizers," IEEE Transactions on Evolutionary Computation, vol. 20, no.2, pp. 180-198, doi: 10.1109/TEVC.2015.2443001

Zainol Abidin et al. (2014). Zainol Abidin, H., Din, N.M., Yassin, I.M., H. A., Omar: Sensor Node Placement in Wireless Sensor Network Using Multiobjective Territorial Predator Scent Marking Algorithm. Arab J Sci Eng, 39 (8), 6317-6325 (2014). doi:10.1007/s13369-014-1292-3

Zhang (2003). Zhang JHH. Maintaining sensing coverage and connectivity in large sensor networks. Technical report UIUCDCS-R-2003-2351; 2003

Zhang et al. (2010). Zhang C, Bai X, Teng J, Xuan D, Jia W. Constructing low-connectivity and full-coverage three dimensional sensor networks. IEEE Journal on Selected Areas in Communications 2010;28(7):984–93

Zhang and Li. (2007). Zhang, Q. and Li, H., "MOEA/D: A Multiobjective Evolutionary Algorithm Based on De-composition", IEEE Transactions on Evolutionary Computation, 11, 6, pp. 712-731. Doi: 10.1109/TEVC.2007.892759

Zhang et al. (2018). Zhang H, Liu Y, Zhou J. Balanced-evolution genetic algorithm for combinatorial optimization problems: the general outline and implementation of balanced evolution strategy based on linear diversity index. Nat Comput 1-29. https://doi.org/10.1007/s11047-018-9670-5

Zhang et al. (2014). Z. Zhang, K. Long, J. Wang, F. Dressler. On Swarm Intelligence Inspired Self-Organized Networking: Its Bionic Mechanisms, Designing Principles and Optimization Approaches. IEEE Communications Surveys and Tutorials, 16(1):513-537

Zhang et al. (2004). P. Zhang, C.M. Sadler, S.A. Lyon, and M. Martonosi, "Hardware design experiences in ZebraNet," in Proceedings of the SenSys'04, Baltimore, MD, 2004

Zhang et al. (2006). L. Zhang, X. Zhou, and Q. Cheng, "Landscape-3D: a robust localization scheme for sensor networks over complex 3D terrains," In Proceedings of the 31st Annual IEEE Conference on Local Computer Networks (LCN '06), pp. 239–246.

Zitzler et al. (1999). Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Trans Evol Comput 3(4):257–271. https://doi.org/10.1109/4235.797969

Appendix 1
RSSI and FER measures of in our experimental tests (Scenario 1)

Table A1 Averages of RSSI values by neighbours of 'C' in different positions during day

\		N1	N2	N3	N4	N5	N6	N7	N8	N9	N10 (A)	N11 (B)	VMR(Initial deployment)	VMR(∈-NSGA-II-AxN-AmN)	VMR(U-NSGA-III-AxN-AmN)	VMR(mMOEA/DD-AxN-AmN)
P1	RSSI generated by 'C'				130								163	158,5	163,2	166,5
	RSSI detected by 'C'	79	96	81	163	88		75	75	81	88	88				
P2	RSSI generated by 'C'		137		135								128	132,8	134,2	138,2
	RSSI detected by 'C'	106	105	77	168	101	78	91	99	77	97	80				
P3	RSSI generated by 'C'	128		128	116								123,5	124,9	132,7	139,4
	RSSI detected by 'C'	101	91	109	132	79	78	88	105	86	80	87				
P4	RSSI generated by 'C'	151		133	115				129				125,37	127,6	138,5	140,1
	RSSI detected by 'C'	142	95	125	144	118	105	97	139	101	112	88				
P5	RSSI generated by 'C'				111				130				120	124,8	126,6	129,6
	RSSI detected by 'C'	91	89	94	134	104	108	82	138	93	104	88				
P6	RSSI generated by 'C'				109				128				128,25	131,5	128,8	134,5
	RSSI detected by 'C'	93	90	94	136	117	98	74	141	71	119					
P7	RSSI generated by 'C'				113				129				123,4	125,57	129	132,8
	RSSI detected by 'C'	93	87	84	141	120	107	80	133	81	116					
P8	RSSI generated by 'C'				109								112,5	119,5	119,5	125,6
	RSSI detected by 'C'	77	88	91	129	102	91		113	83	106					
P9	RSSI generated by 'C'												106	117	121	124,5
	RSSI detected by 'C'	78	83	89	106	99	83		89		89					
P10	RSSI generated by 'C'												119,33	123	126,3	129,6
	RSSI detected by 'C'	93	86	98	97	125			120		113					
P11	RSSI generated by 'C'		138		109						130		119,66	124,14	127,8	133,3
	RSSI detected by 'C'	75	76	99	129	123	98	101	74		124	103				
P12	RSSI generated by 'C'				134								129	133	131	133,4
	RSSI detected by 'C'	93	91	95	146	99	90	92	112	88	95	90				
P13	RSSI generated by 'C'				120								124,66	132,2	130,5	132,2
	RSSI detected by 'C'	87	91	96	153	99	93	115	99		106	81				
P14	RSSI generated by 'C'				131								130	127,6	129,9	134,9
	RSSI detected by 'C'	81	83	85	173	86		113	98	84	93	104				
P15	RSSI generated by 'C'				122			124					121	127	127	128,8
	RSSI detected by 'C'	94	83	111	135	79		123		97		98				
P16	RSSI generated by 'C'				128								128	131	133,2	133,8
	RSSI detected by 'C'	91	87	84	112	93		98	79	84	84	89				
P17	RSSI generated by 'C'												106	107,5	107,5	112,3
	RSSI detected by 'C'		80	98	106	79		87	88		89	82				
Average values													123,98	127,506	129,805	133,5

* **VMR (*𝒜*)**: Let ***Vi*** be the upper value between the RSSI value of detecting 'C' by the node Ni and the RSSI value of detecting Ni by the node 'C' in a given position Pi considering the new nomad node's positions given by the algorithm *𝒜*. **VMR***(𝒜)* is the average value of the RSSI, it is defined by dividing the sum of ***Vi*** by the total number of nodes detected by or detecting the node 'C'.

* The positions of nomad nodes and their corresponding values of RSSI in each location Pi are not mentioned in Tables A1 and A2. RSSI values in Table A1 and Table A2 are expressed in values between 0 and 255, not in dBm.

Table A2 Averages of RSSI values by neighbours of 'C' in different positions overnight.

\s		N1	N2	N3	N4	N5	N6	N7	N8	N9	N10 (A)	N11 (B)	VMR(Initial deployment)	VMR(ϵ-NSGA-II-AxN-AmN)	VMR(U-NSGA-III-AxN-AmN)	VMR(mMOEA/DD-AxN-AmN)
P1	RSSI generated by 'C'				135								120,5	116,4	118,9	126,7
	RSSI detected by 'C'	89	106	104	170	102	91	91	98	79	99	93				
P2	RSSI generated by 'C'	129	129		134								121,125	121,125	121	129,5
	RSSI detected by 'C'	107	104	103	169	104		93	118	100	117	98				
P3	RSSI generated by 'C'	131	132	130	109								133,4	134,28	131,4	139,2
	RSSI detected by 'C'	104	118	120	156	85	81	89	118	90	94	81				
P4	RSSI generated by 'C'	128											114.5	114,5	115	119,8
	RSSI detected by 'C'	118	87	101	123	93	90	82	106	82	82	81				
P5	RSSI generated by 'C'	129		137	118								125.5	126,09	128,4	132,9
	RSSI detected by 'C'	117	110	125	148	112	102	81	150	87	116	97				
P6	RSSI generated by 'C'												106	108,66	107,2	111,8
	RSSI detected by 'C'	76	82	68	110	76	85		102		77	72				
P7	RSSI generated by 'C'					136							117,2	117,33	121,6	126,3
	RSSI detected by 'C'	95	106	82	111	128	92	75	124	77	109					
P8	RSSI generated by 'C'												112	114,5	118,8	124,4
	RSSI detected by 'C'	73	82	79	118	106	83	75	98	77	98					
P9	RSSI generated by 'C'												105	107,75	110,7	116,6
	RSSI detected by 'C'	92	76	75	100	96	80		110	72	98					
P10	RSSI generated by 'C'					134							117,66	122,5	125,6	133,3
	RSSI detected by 'C'	73	85	79	93	123	71		107	76	112					
P11	RSSI generated by 'C'					150							128	137,6	140,3	145,5
	RSSI detected by 'C'	75	95	99	119	118			97		115					
P12	RSSI generated by 'C'												113	115.7	113,8	119,8
	RSSI detected by 'C'	85	87	74	123	74	76	86	103	72	78	78				
P13	RSSI generated by 'C'				123								110,25	120,28	112,5	118,5
	RSSI detected by 'C'		31	90	118	104	92	85	111	90	108	82				
P14	RSSI generated by 'C'				157								118,66	128,75	133,4	131,9
	RSSI detected by 'C'	102	80	102	185	74		106	111	94	79	106				
P15	RSSI generated by 'C'				137			143					130,8	136,85	139,8	144,2
	RSSI detected by 'C'	90	74	106	169	80		129	77	119	77	117				
P16	RSSI generated by 'C'												118	121,33	126,7	129,6
	RSSI detected by 'C'	73	76	72	118	88		81	72	73	73	86				
P17	RSSI generated by 'C'												118	120	120,8	127,8
	RSSI detected by 'C'	73	75	74	118	91		79	79	82	94	79				
Average values													117,973	121,746	122,7	128,105

Table A3 Averages of FER values by neighbours of 'C' in different positions during day

\		N1	N2	N3	N4	N5	N6	N7	N8	N9	N10 (A)	N11 (B)	VMF (Initial deployment)	VMR(ϵ-NSGA-II-AxN-AmN)	VMR(U-NSGA-III-AxN-AmN)	VMR(mMOEA/DD -AxN-AmN)
P1	FER generated by 'C'				0,244								0,1287	0,0878	0,0868	0,0785
	FER detected by 'C'	**0,661**	**0,037**	**0,459**	0,013	**0,684**		**0,684**	**0,49**	**0,581**	**0,105**	**0,045**				
P2	FER generated by 'C'		0		0,115								0,35392	0,31194	0,3	0,28
	FER detected by 'C'	0,356	0,171	**0,196**	0,006	0,144	**0,8**	**0,464**	**0,168**	**0,891**	**0,21**	**0,74**				
P3	FER generated by 'C'	0,507		0,292	0,929								0,32305	0,32305	0,301	0,268
	FER detected by 'C'	0,048	**0,1**	**0,7**	0,006	**0,057**	**0,972**	**0,121**	0,051	**0,784**	**0,141**	**0,847**				
P4	FER generated by 'C'	0,051		0,203	0,857				0,872				0,21201	0,17811	0,145	0,102
	FER detected by 'C'	0,060	**0,855**	0,097	0,082	0,049	0,126	**0,877**	0,052	0,314	0,069	**0,333**				
P5	FER generated by 'C'				0,914				0,352				0,257425	0,22334	0,199	0,168
	FER detected by 'C'	**0,218**	**0,208**	**0,260**	0,028	0,198	**0,553**	**0,912**	0,024	**0,31**	0,172	**0,936**				
P6	FER generated by 'C'				0,977				0,913				0,379925	0,261708	0,2734	0,23481
	FER detected by 'C'	**0,043**	**0,66**	**0,132**	0,016	0,030	**0,224**	**0,956**	0,017	**0**	0,031					
P7	FER generated by 'C'				0,968				0,951				0,20183	0,216775	0,18	0,12
	FER detected by 'C'	**0,202**	**0,215**	**0,5**	0,009	0,017	0,108	**0**	0,009	**0,666**	0,023					
P8	FER generated by 'C'				0,948								0,129375	0,1048333	0,164	0,092
	FER detected by 'C'	**0,452**	**0,236**	**0,522**	0,008	0,015	**0,175**		0,007	**0,744**	0,016					
P9	FER generated by 'C'												0,0096	0,0087	0,0139	0,0084
	FER detected by 'C'	**0,8**	**0,59**	**0,866**	0,009	**0,017**	**0**		**0,052**		**0,042**					
P10	FER generated by 'C'												0,1097	0,09468	0,078	0,083

Appendix 1. RSSI and FER measures of in our experimental tests (Scenario 1)

	FER detected by 'C'	0,875	0,718	0,732	0,067	0,017			0,213		0,099					
P11	FER generated by 'C'		0,733		0,912						0,8		0,3218	0,2895214	0,232	0,198
	FER detected by 'C'	0,75	0,52	0,861	0,627	0,014	0,769	0	0,065		0,028	0				
P12	FER generated by 'C'				0,082								0,2191	0,186125	0,18956	0,136
	FER detected by 'C'	0,625	0,298	0,465	0,219	0,080	0,64	0,72	0,017	0	0,055	0,769				
P13	FER generated by 'C'				0,708								0,3545	0,234	0,249	0,202
	FER detected by 'C'	0,551	0,574	0,677	0,017	0,554	0	0,363	0,038		0,338	0,87				
P14	FER generated by 'C'				0,129								0,2067	0,18885	0,211	0,074
	FER detected by 'C'	0,826	0,918	0,421	0,011	0,441		0,27	0,263	0,819	0,139	0,28				
P15	FER generated by 'C'				0,666			0,013					0,1354	0,109075	0,0798	0,037
	FER detected by 'C'	0,058	0,937	0,054	0,004	0		0,021		0,056		0,017				
P16	FER generated by 'C'				0,245								0,1685	0,11675	0,089	0,053
	FER detected by 'C'	0,4	0,311	0,648	0,092	0,25		0,05	0,66	0,028	0,615	0,03				
P17	FER generated by 'C'												0,072	0,0595	0,0598	0,059
	FER detected by 'C'		0,135	0,685	0,072	0,326		0,030	0,666		0,280	0,917				
Average values													0,21079	0,17616	0,16772	0,12904

* **VMF ($\mathcal{A}$)**: Let ***Vi*** be the average value between the FER value of detecting 'C' by the node 'Ni' and the FER value of detecting 'Ni' by 'C' in a given position Pi considering the new nomad node's positions given by the algorithm $\mathcal{A}$. **VMF($\mathcal{A}$)** is the average value of the FER. It is defined by dividing the sum of the ***Vi*** values by the total number of nodes detecting or detected by 'C'.

* The positions of nomad nodes and their corresponding values of FER in each location Pi are not mentioned in Tables A3 and A4.

Table A4 Averages of FER values by neighbours of 'C' in different positions overnight

\		N1	N2	N3	N4	N5	N6	N7	N8	N9	N10 (A)	N11 (B)	VMF (Initial deployment)	VMR(ϵ-NSGA-II-AxN-AmN)	VMR(U-NSGA-III-AxN-AmN)	VMR(mMOEA/DD -AxN-AmN)
P1	FER generated by 'C'				0,048								0,1675	0,1469	0,133	0,128
	FER detected by 'C'	0,439	0,268	0,156	0,008	0,218	0	0,156	0,159	0,091	0,294	0,08				
P2	FER generated by 'C'	0,185	0,182		0,036								0,1128	0,1122	0,105	0,096
	FER detected by 'C'	0,198	0,01	0,023	0,005	0,040		0,464	0,022	0,451	0,059	0,194				
P3	FER generated by 'C'	0,265	0,25	0,154	0,542								0,1394	0,1092	0,102	0,094
	FER detected by 'C'	0,019	0,091	0,016	0,005	0,06	0,097	0,589	0,026	0,628	0,056	0,099				
P4	FER generated by 'C'	0,258											0,0573	0,0522	0,044	0,033
	FER detected by 'C'	0,029	0,446	0,022	0,006	0,734	0,781	0,615	0,058	0,246	0,416	0,775				
P5	FER generated by 'C'	0,932		0,162	0,853								0,2612	0,1988	0,186	0,159
	FER detected by 'C'	0,202	0,181	0,261	0,032	0,105	0,345	0,813	0,055	0,623	0,183	0,666				
P6	FER generated by 'C'												0,104	0,074	0,0114	0,0028
	FER detected by 'C'	0,821	0,333	0,083	0,205	0,066	0,101		0,003		0,693	0				
P7	FER generated by 'C'					0,498							0,1319	0,1174	0,129	0,098
	FER detected by 'C'	0,166	0,165	0,68	0,009	0,023	0,040	0	0,005	0	0,220					
P8	FER generated by 'C'												0,1105	0,0768	0,088	0,076
	FER detected by 'C'	0,617	0,644	0,79	0,211	0,010	0,705	0,936	0,010	0,901	0,021					
P9	FER generated by 'C'												0,0295	0,0122	0,0326	0,008
	FER detected by 'C'	0,736	0,607	0,846	0,020	0,022	0,5		0,039	0,930	0,031					
P10	FER generated					0,046							0,0649	0,054	0,061	0,052

Appendix 1. RSSI and FER measures of in our experimental tests (Scenario 1)

	by 'C'															
	FER detected by 'C'	**0,710**	**0,111**	**0,200**	**0,037**	**0,023**	**0,923**		**0,120**	**0**	**0,040**					
P11	FER generated by 'C'					**0,276**							0,2954	0,2383	0,266	0,241
	FER detected by 'C'	**0,95**	**0,118**	**0,954**	**0,030**	**0,074**			**0,690**		**0,681**					
P12	FER generated by 'C'												0,5315	0,3024	0,4036	0,29
	FER detected by 'C'	**0,863**	**0,627**	**0,902**	**0,643**	**0,7**	**0,636**	**0,631**	**0,39**	**0,828**	**0,307**	**0,425**				
P13	FER generated by 'C'				**0,775**								0,1706	0,1512	0,166	0,147
	FER detected by 'C'		**0,795**	**0**	**0,007**	**0,079**	**0,714**	**0**	**0,12**	**0,826**	**0,092**	**0,166**				
P14	FER generated by 'C'				**0,082**								0,1293	0,1163	0,128	0,115
	FER detected by 'C'	**0,075**	**0,765**	**0,089**	**0,005**	**0,125**		**0,012**	**0,018**	**0,065**	**0,653**	**0,538**				
P15	FER generated by 'C'				**0,121**			**0**					0,0287	0,0249	0,0251	0,0242
	FER detected by 'C'	**0,155**	**0,757**	**0,031**	**0,004**	**0,183**		**0,013**	**0,121**	**0,024**	**0,898**	**0,015**				
P16	FER generated by 'C'												0,29	0,196	0,188	0,167
	FER detected by 'C'	**0,869**	**0,615**	**0,827**	**0, 29**	**0,284**		**0,294**	**0,555**	**0,732**	**0,86**	**0,232**				
P17	FER generated by 'C'												0,0534	0,0279	0,0252	0,018
	FER detected by 'C'	**0**	**0,688**	**0,901**	**0,053**	**0,315**		**0,194**	**0,74**	**0,975**	**0,393**	**0,294**				
Average values													0,15125	0,11827	0,12317	0,10288

Appendix 2

Résumé Long en Français
Contributions au déploiement optimisé des capteurs connectés dans les réseaux de collecte de l'Internet des Objets

Mots-clefs : Réseau de collecte de l'IoT, optimisation, redéploiement 3D d'intérieur, validation expérimentale, méta-heuristique, hybridation, algorithme de colonies de fourmis, algorithmes génétiques, optimisation par essaim particulaire, réduction de dimensionnalité, incorporation de préférences.

1 Motivations et problématique

Pour différents contextes d'application, le domaine des réseaux de capteurs sans fil (RCSF) est un axe de recherche en évolution continue, surtout avec l'émergence de l'internet des objets (IoT). Par la suite, l'IoT est un concept qui se relie étroitement aux problématiques traitées dans notre étude. L'IoT offre un scénario dans lequel différents entités hétérogènes et communiquant appelées objets/choses (personnes, robots ou appareils) sont connectées et distinguées par des identifiants uniques. Ces entités peuvent transférer des données sur le réseau automatiquement sans intervention humaine. Le RCSF est le pont reliant le monde numérique au monde réel. Il permet d'assurer la communication matérielle pour transmettre et récupérer les valeurs réelles détectées par les objets connectées sans fil (nœuds de capteurs). Le rôle de l'IoT est celui de traiter ces données, de les manipuler et de prendre les décisions. Dans nos travaux de recherche, nous nous intéressons aux réseaux DL-IoT (Device Layer-IoT) qui représentent le réseaux de collecte utilisé pour obtenir des données à partir des nœuds de capteurs réparties dans l'environnement réseau. De ce fait, notre approche peut etre appliquée pour les contextes WSN et IoT. A cet égard, face à la capacité limitée en énergie, en traitement et en mémoire des capteurs/objets, le DL-IoT soulève de nombreux problèmes d'optimisation. Notre contribution globale s'oriente vers la proposition des heuristiques, des méta-heuristiques hybrides (centralisées et distribuées), de modèles multi-objectif, de formulations mathématiques et d'évaluations de récents approches d'optimisation sur le DL-IoT. Compte tenu de la croissance continue dans les thématiques du DL-IoT, différents contextes peuvent être investiguées tel que l'optimisation du déploiement des nœuds.
Le déploiement des nœuds représente la première phase dans l'installation d'un WSN. En termes d'optimisation de consommation d'énergie, ceci influe considérablement sur la performance, la fiabilité et le fonctionnement du réseau. On peut décrire le problème de déploiement des nœuds comme étant le positionnement de l'ensemble des capteurs constituant le réseau et se trouvant disséminés de manière aléatoire. Pour compenser le caractère erratique de leur placement, on déploie souvent un grand nombre de capteurs intelligemment. Ceci peut contribuer également à l'augmentation de la tolérance aux fautes du réseau. Pour le déploiement 2D de réseaux de capteurs sans fil, une littérature très abondante existe allant de l'organisation topologique du réseau aux spécificités techniques des capteurs et leur manière de communiquer. Pourtant, le déploiement 3D à moindre coût pose de très nombreux problèmes d'optimisation avec une littérature peu abondante. Donc, notre objectif est de trouver l'architecture la plus optimisée et de fournir des solutions d'organisation 3D des capteurs, tout en améliorant la performance du réseau. A cet égard, plusieurs issues et objectifs sont reliées au problème de déploiement des réseaux de capteurs tels que la consommation d'énergie, la durée de vie et la localisation. De ce fait, différents approches heuristiques pour l'optimisation dans les WSN peuvent être envisagées.

La nature distribuée et dynamique du problème de déploiement exige l'utilisation de méthodologies avancées d'optimisation et de modélisation : les méta-heuristiques hybrides. Ces dernières permettent d'exploiter la complémentarité de ces méthodes entre elles, et de tirer les avantages des autres approches classiques hybrides avec eux. Face à nos objectifs qui concernent principalement l'assurance d'un système évolutif, flexible et robuste contre les perturbations du réseau, cette nouvelle classe de méta-heuristiques hybrides a montré sa performance dans la résolution des problèmes d'optimisation difficiles. Notre contribution consiste à proposer un ensemble de modifications et hybridations bien justifiées sur les méta-heuristiques, tout en les appliquant et évaluant leurs performances sur notre problème de déploiement issu du monde réel. A cet égard, les problèmes d'optimisation les plus difficiles à résoudre sont généralement dérivés de problématiques réelles qui sont souvent non mesurables, complexes et antagonistes. Ils sont, majoritairement, des problèmes comprenant plusieurs critères. Ceci est dû au fait qu'ils possèdent plusieurs objectifs d'évaluations, souvent contradictoires et à considérer simultanément. Ceci a donné naissance à la théorie de l'optimisation multi-objectif. Dans la littérature, plusieurs méthodes de résolution des problèmes multi-objectifs ont été développées. On peut classifier ces méthodes en deux grandes classes : les méthodes exactes et les méthodes approchées qui sont subdivisées en des heuristiques spécifiques et des méta-heuristiques. Les méta-heuristiques forment un ensemble d'algorithmes d'optimisation visant à résoudre les problèmes d'optimisation difficiles. Ces algorithmes permettent d'améliorer la qualité des solutions sans garantir l'optimalité de la solution obtenue mais dans un temps de calcul très raisonnable par rapport à la complexité du problème souvent insolvable dans un temps polynomial. Les problèmes d'optimisation du monde réel ont souvent plusieurs objectifs et contraintes conflictuels et contradictoires (nommés problèmes d'optimisation multi-objectif (MOP)). Ceci implique qu'il n'existe pas une solution unique qui est optimale par rapport à tous ces objectifs simultanément. La solution est en général plusieurs solutions non comparables, et l'ensemble des compromis de ces solutions est appelé « front de Pareto » (PF). Le but du MOP est de trouver une approximation bien répartie et bien convergente du PF. Par la suite, le décideur (DM) peut sélectionner la solution préférée. Pour déterminer le PF, différents méthodes, appelés algorithmes évolutionnaires (EA) ont été proposées dans la littérature en se basant sur l'imitation des principes de l'évolution biologique. Les EA sont devenus populaires et largement utilisés dans la résolution des MOP à cause de leurs insensibilité aux aspects et formes des fonctions objectives (Deb 2001) telles que la multi-modalité, la discontinuité, la convexité et l'uniformité de l'espace de recherche, etc.. En effet, l'optimisation multi-objectif évolutionnaire (EMO) est une récente branche de l'optimisation qui est apparue suite à ce succès des algorithmes d'optimisation multi-objective évolutionnaires (MOEA) dans la résolution MOP.

2 Organisation du document

Ce rapport comporte quatres chapitres répartis sur deux parties. Une première partie qui concerne l'état d'art et comporte deux premiers chapitres ; et une deuxième partie qui concerne les contributions proposées et qui se compose des chapitres 3 et 4. Nous commençons dans le premier chapitre par présenter l'état de l'art du problème de déploiement 3D indoor. Nous introduisons les défis de déploiement tridimensionnel et ses différents types, objectives, modèles et applications. Ensuite, des travaux de recherche récents portant sur le problème du déploiement ont été identifiés et critiqués. Le deuxième chapitre permet d'introduire les méthodes utilisées pour résoudre le déploiement en 3D à l'intérieur, en particulier les algorithmes d'optimisation évolutive, les systèmes multi-agents et les concepts d'incorporation de la réduction et des préférences de l'utilisateur. Nous détaillons, également, les concepts fondamentaux d'optimisation multi-objectifs tels que la domination, la frontière de Pareto. Nous décrivons aussi les approches principales et méta-heuristiques classiques de la résolution pour ces problèmes multi-objectifs. Dans le troisième chapitre, nous présentons

une modélisation mathématique du problème de déploiement, basée sur une formulation de programmation linéaire entière. Nous présentons alors l'architecture des nœuds, les hypothèses, la notation et la fonction objective. Ensuite, nous proposons les objectifs et leurs spécificités. Ce troisième chapitre est aussi consacré à expliquer et justifier les hybridations et modifications introduites aux méta-heuristiques, dans l'optique d'améliorer leurs performances et capabilités à résoudre les problèmes d'optimisation. En particulier, nous détaillons les problèmes complexes et à plusieurs objectifs du monde réel tel que le nôtre. Ces modifications concernent essentiellement l'utilisation du voisinage et d'un concept adaptatif pour les opérateurs de mutation et de recombinaison, l'utilisation de fonctions de scalarisation multiples dans l'approche basée sur l'agrégation. Nous détaillons l'incorporation de la réduction de dimensionnalité, l'incorporation des préférences des utilisateurs, un cadre hybride pour NSGA-III et colonies de fourmis, enfin la proposition d'un concept d'accents d'oiseaux dans PSO et son hybridation avec les systèmes multi-agents. Le dernier (quatrième) chapitre est consacré à l'illustration des résultats des différents tests expérimentaux. Nous commençons par introduire les différents paramètres de test et les métriques d'évaluation que nous avons utilisée pour valider nos propositions. Nous passons par la suite à l'illustration des résultats numériques du problème de déploiement 3D après avoir incorporé notre schéma proposé de réduction de dimensionnalité et de préférences. Ensuite, nous détaillons les résultats sur les fonctions de test sans contrainte DTLZ. Une étude qui présente la modélisation des protocoles utilisés puis des simulations avec des instances petites et grandes. Ensuite, nous proposons des expériences sur des bancs d'essai réels et les résultats obtenus lors de l'hybridation d'ACO et de NSGA-III, puis les résultats de l'algorithme PSO basé sur le concept d'accents d'oiseaux et l'algorithme d'hybridation PSO et SMA. Enfin, nous présentons nos conclusions ainsi que nos perspectives et pistes de recherches futures.

3 Etude bibliographique et principales contributions

Le but de cette thèse est de proposer des hybridations et modifications des algorithmes d'optimisation évolutionnaire dans le but de réaliser le positionnement adéquat des nœuds dans les réseaux de capteurs sans fil avec satisfaction d'un ensemble de contraintes et objectifs.

3.1 Étude bibliographique

En première étape on a réalisé une étude bibliographique approfondie portant sur les méthodes d'optimisation du déploiement dans les réseaux de capteurs sans fil, particulièrement le déploiement 3D à l'intérieur.

3.1.1 Travaux en relation avec la problématique de déploiement 2D-3D

Plusieurs travaux récents ont proposé des algorithmes d'optimisation évolutifs pour un déploiement efficace des nœuds WSN. (Banimelhem et al., 2013) proposent un algorithme génétique (AG) pour résoudre le problème des trous de couverture et du déploiement 2D déterministe dans les WSN. L'objectif est de minimiser le nombre de nœuds mobiles. Cependant, aucune modélisation mathématique n'est donnée au problème. (Danping et al., 2013) proposent un algorithme évolutif multi-objectif combiné à une heuristique à bas coût pour résoudre le problème du signal et de la propagation radio en intérieur avec un scénario de déploiement en 3D. L'objectif est d'optimiser la couverture, la qualité de la liaison, la durée de vie et le coût du matériel. Cependant, l'évolutivité de l'approche proposée n'est pas prouvée. Ko and Gagnon (2015) suggèrent un AG reposant sur un schéma d'analyse du croisement pour résoudre le déploiement dans des terrains 3D irréguliers. L'objectif est de maximiser la couverture probabiliste des points et la couverture globale. En revanche, ils n'ont pas prouvé que la stratégie de croisement proposée est meilleure que l'approche génétique originale. Alia and Al-Ajouri (2017) proposent un algorithme basé sur une recherche d'harmonie

pour optimiser le nombre de capteurs déployés et la couverture. L'inconvénient de ce travail est que le modèle de réseau proposé est simpliste. De plus, seuls deux objectifs sont considérés et la validation de l'approche est basée uniquement sur des tests Matlab, sans scénario de simulation ou d'expérimentation du monde réel.

3.1.2 Travaux en relation avec l'hybridation par les procédures de préférence des utilisateurs et la réduction de dimensionnalité

De nombreuses recherches récentes ont suggéré des approches évolutives, telles que les algorithmes génétiques (AG) pour garantir un déploiement efficace dans les RCSF. Cependant, cette étude est basée sur une simple comparaison entre les deux algorithmes mentionnés et les évaluations ne s'appuient que sur des simulations sans validations expérimentales. (Unaldi et al., 2017) ont proposé un algorithme basé sur une mutation aléatoire et une transformée d'ondelette guidée pour résoudre le problème du déploiement probabiliste de nœuds dans des terrains 3D. L'objectif est de maximiser la qualité de la couverture et de minimiser le nombre de capteurs. Cependant, la méthode proposée est évaluée à l'aide de capteurs stationnaires et aucun environnement dynamique n'est utilisé. De plus, cette étude n'est pas validée par des scénarios empiriques sur un problème du monde réel. En outre, des simulations sont réalisées sans utiliser des mesures d'évaluation connues telles que l'IGD (Coello et al 2002) ou l'Hypervolume (Zitzler al., 1999) pour évaluer la performance de l'approche. Concernant les algorithmes évolutifs, Zhang et al. (2018) ont proposé d'injecter le hasard dans la population pour mieux contrôler la diversité. Ils ont proposé un algorithme génétique appelé (BEGA) basé sur une matrice de similarité (SGM). Bien que BEGA soit comparé à différents autres variantes génétiques pour douze benchmarks, aucun test sur des problèmes industriels réels n'est proposé et aucune preuve d'efficacité de son application dans les problèmes à grande échelle n'est donnée. En ce qui concerne l'incorporation des préférences de l'utilisateur dans des algorithmes d'optimisation à objectifs multiples, un algorithme appelé *Prio- ε-Preferred* basé sur une relation *ε-preferred* est proposé par (Drechsler et al., 2015). Cependant, pour prouver l'efficacité de l'algorithme suggéré, il est comparé uniquement à l'algorithme standard NSGA-II et il n'y a aucune comparaison avec d'autres approches de préférences de l'utilisateur.

3.1.3 Travaux en relation avec l'hybridation des PSO et SMA

Différentes études suggèrent divers schémas pour hybrider l'OEP (Optimisation par Essaim Particulaires) et les SMA (Systèmes multi-agents). Kolomvatsos and Hadjieftymiades (2014) proposent dans un algorithme OEP reposant sur des négociations intelligentes et simultanées entre un agent acheteur et un agent vendeur pour réaliser une transaction d'achat. L'algorithme OEP a été exécuté sur chaque thread avec une stratégie d'intelligence de l'essaim pour obtenir l'accord optimal. Cependant, seulement deux objectifs sont considérés dans cette étude. Dans (Wu el al., 2015), un système multi-agent basé sur une approche d'optimisation par l'essaim de particules (MAPSO) est utilisé pour résoudre le problème de la distribution de la charge économique des systèmes d'alimentation. En effet, chaque agent coopère avec ses voisins pour ajuster sa capacité de recherche globale et sa capacité d'exploration locale. Bien que les résultats indiquent que cet algorithme a une grande précision et vitesse de convergence par rapport aux autres algorithmes évolutionnaires, cet algorithme doit être testé sur un problème du monde réel avec un plus grand nombre d'objectifs et une dimension du problème plus élevée. De plus, la modélisation du problème n'est pas présentée. Dans (Ahmad et al., 2007), les auteurs proposent un autre algorithme OEP basé sur un SMA. Avec un degré d'apprentissage plus élevé et une exécution asynchrone, les particules de cet algorithme ont plus d'autonomie que les particules dans l'OEP standard. L'environnement de ces particules d'agent est modélisé comme un groupe de points non-optimaux. Les auteurs ont également mis en œuvre une variante parallèle de l'algorithme proposé, qui est plus efficace mais ayant

une complexité algorithmique plus élevée. Une autre étude qui combine l'OEP, les SMA et les AG a été proposée dans (Peng et al., 2008). Les auteurs suggèrent l'intégration des approches mentionnées pour résoudre le problème de la génération automatique de papier de test dans un environnement informatique parallèle. L'architecture multi-agent proposée est basée sur un nœud distant, un nœud central et un ensemble d'agents appelés *TPAgents* qui contrôlent les opérations d'évolution de chaque génération de population dans l'AG. Pour minimiser le coût de la communication entre les nœuds, une évaluation de la condition physique est effectuée par *TPAgents* au niveau des nœuds locaux. Seule la meilleure particule locale et sa fonction d'adaptation (*fitness*, en anglais) locale correspondante sont envoyées au nœud central pour calculer la meilleure particule globale. Pour échapper aux optimum locaux, la meilleure particule globale est transmise aléatoirement au nœud distant. Bien que l'approche proposée soit efficace, son évolutivité n'est pas prouvée et le nombre d'agents utilisés est assez faible. De même, aucun déploiement d'application réelle basée sur un système prototype n'est proposé. Dans (Kumar et al., 2011), un algorithme d'optimisation d'essaim de particules multi-agent hybridé avec un processus de prise de décision d'abeilles est proposé pour résoudre le problème de répartition de puissance. L'avantage de cette étude est que l'algorithme proposé est performant avec différentes fonctions objectives et sur des problèmes d'optimisation non contraints. Cependant, le temps de convergence de cet algorithme augmente considérablement si le nombre d'agents augmente.

Tous les travaux mentionnés ci-dessus utilisent des schémas d'hybridation simplistes basés sur l'idée de modéliser la particule en tant qu'agent. La modélisation environnementale, la capacité des particules et les mécanismes d'interaction entre les agents sont souvent non spécifiés. Dans notre approche, nous visons à proposer un schéma d'hybridation plus sophistiqué des deux approches (OEP et SMA) en utilisant le nouveau concept d'accent.

3.2 Formulation mathématique

Dans la deuxième étape, une formulation mathématique vise la modélisation de la problématique. Il s'agit de fixer la fonction objective à optimiser, les variables de décision et les différentes contraintes à prendre en considération. Notre objectif est de minimiser le nombre de nœuds capteurs à utiliser et la consommation de l'énergie. Simultanément, on vise ainsi la maximisation de la durée de vie du réseau, mais aussi la couverture, la localisation et la connectivité. Une modélisation détaillée peut être consultée dans le troisième chapitre de ce rapport de thèse. Dans ce qui suit on explique nos propositions de différents schémas d'hybridation justifiés et de modifications entre les algorithmes d'optimisation pour mieux résoudre notre problématique.

3.3 Inclusion de la diversité

Dans une première approche, une mutation adaptative et des opérateurs de recombinaison avec des contraintes de croisement avec le voisinage sont proposées. L'utilisation d'un concept de fonctions de scalarisation multiples est introduit pour faire face : (i) à l'inefficacité des Algorithmes évolutionnaires multi-objectives à base de Pareto, (ii) à l'inefficacité de l'opération de recombinaison, et (iii) à l'augmentation exponentielle des coûts (temps et espace) lors de la résolution de problèmes à plusieurs objectifs dans le contexte du monde réel. Pour cela on propose des mécanismes d'inclusion de diversité qui se basent essentiellement sur les concepts suivants (pour plus de détails, consultez la section 3.3.2 du rapport):

Concept de recombinaison avec le voisinage : Dans le cas des problèmes MaOP (ayant un nombre d'objectives supérieur à trois), la dimension de l'espace objectif devient trop élevée, ce qui augmente la diversité de la population. Ainsi, les opérateurs de mutation et de

recombinaison deviennent inefficaces. D'où la possibilité qu'ils créent des individus qui ne sont pas sélectionnés comme parents. Par la suite, les algorithmes MaOA résolvant ce genre de problèmes deviennent plus efficaces si la recombinaison est effectuée avec un chromosome voisin et si les objectifs sont corrélés. En effet, le concept de voisinage utilisé repose sur les étapes suivantes:
- le calcul de la distance séparant les individus dans l'espace objectif,
- l'identification du sous-ensemble de ($|P| .N_s$) voisins les plus proches pour chaque individu, où N_s est la taille du voisinage et P est la population. D'après QU (2013), N_s entre 1/20 et 1/5 est préféré. Dans nos tests, on a fixé N_s à 1/10.

Concept de multi-opérateurs adaptatifs : lors de la résolution des problèmes MaOP, les MaOA sont souvent incapables de trouver les opérateurs de mutation et de recombinaison appropriés en fonction des spécificités du problème à résoudre. Pour surmonter cette faiblesse, nous suggérons de faire varier l'opérateur utilisé de manière adaptative. En effet, la contribution d'un opérateur à l'itération précédente est prise en compte pour l'ajustement de sa probabilité d'être sélectionné lors des itérations suivantes. Par ailleurs, les informations de retour d'expérience des générations passées permettent de sélectionner les opérateurs dans les générations futures sans affecter l'aspect probabiliste des opérateurs.

Résultats de l'utilisation des restrictions d'association de voisinage et d'opérateurs de recombinaison adaptative

Nous examinons l'effet de la stratégie proposée de recombinaison avec le voisinage et de multi-opérateurs adaptatifs. La performance de chaque algorithme est évaluée en utilisant la métrique Hypervolume (HV) (une valeur moyenne du HV sur 15 exécutions. La taille de la population dans MOEA/D est fixée à 1000. La probabilité de mutation est de 1/500 et la probabilité de recombinaison est de 0,8. La recombinaison est basée sur l'appariement avec les voisins et les objectifs sont corrélés. Le nombre de points de référence est fixé à 100. Le tableau B1 illustre les résultats obtenus.

Tableau B1 Valeurs HV optimales, moyennes et pires; en utilisant des opérateurs adaptatifs

	Nbr d'Obj	MOEA/D(PBI)	MOEA/DD	NSGA-III	Two_Arch2
Bit-flip (mutation) / SBX (recombinaison)	4	0.984426	0.983652	0.978863	0.976498
		0.976416	0.981426	0.978574	0.976422
		0.976328	0.976124	0.976231	0.976346
	8	0.969886	0.971841	0.966876	0.952886
		0.969815	0.969822	0.966525	0.952835
		0.969702	0.969723	0.966234	0.952803
Avec restrictions de voisinage et opérateurs adaptives	4	0.978678	0.983129	0.979697	0.977234
		0.978133	0.981952	0.979342	0.976986
		0.976253	0.980237	0.979032	0.976343
	8	0.970489	0.971002	0.971234	0.953864
		0.969932	0.970254	0.967751	0.953366
		0.963231	0.968968	0.963268	0.953032

Les résultats obtenus avec différents nombres d'objectifs indiquent que le croisement avec le voisinage et l'utilisation des opérateurs adaptatifs améliorent considérablement la performance de recherche des solutions. En effet, des meilleurs résultats ont été obtenus pour différents nombres d'objectifs sur l'algorithme MOEA/DD. Lorsque le nombre d'objectifs augmente, l'avantage du MOEA/DD par rapport au MOEA/D devient plus clair. Néanmoins, comparée à d'autres algorithmes, MOEA/D améliore plus significativement la valeur moyenne du HV (avec et sans recombinaison avec des parents similaires). De plus, les résultats expérimentaux montrent que le croisement des parents similaires améliore la diversité sans détériorer la convergence.

3.4 L'incorporation de la réduction de dimensionnalité et de préférences

Dans une deuxième approche, l'incorporation des préférences est établie : en effet, dans le cas des problèmes du monde réel, connus comme ayant un grand nombre d'objectifs, la taille de la population et le nombre de solutions nécessaires dépendent exponentiellement du nombre

d'objectifs. Ainsi, les performances des algorithmes d'optimisation se détériorent lors de la résolution de ce genre de problèmes. Pour résoudre ce défi, l'algorithme NSGA-III est hybridé avec une stratégie interactive d'incorporation des préférences de l'utilisateur (PI-EMO-VF) qui suit l'évolution des nouvelles solutions pour incorporer progressivement les préférences de l'utilisateur. Dans ce travail, l'efficacité du NSGA-III est testée dans le cas de problèmes du monde réel, et comparée à un autre algorithme récent à plusieurs objectifs (MOEA/DD). Dans une troisième approche, un schéma d'hybridation justifié combine trois classes d'algorithmes à objectifs multiples. Ces algorithmes sont basés sur les points de référence (NSGA-III, MOEA/DD), l'agrégation (Two_Arch2) et la décomposition (MOEA/D) avec deux procédures basées sur la réduction de dimensionnalité (MVU-PCA) et les préférences (PI-EMO-PC). Le but de cette hybridation est de bénéficier des avantages de chaque méthode pour résoudre notre problème complexe.

Intégration de PI-EMO-PC, de points Knee et de NL-MVU-PCA sur les EMO testées

Afin de surmonter les difficultés déjà mentionnées des MOA dans la résolution des MaOP, nous proposons un schéma hybride justifié intégrant différentes approches (figure B1).

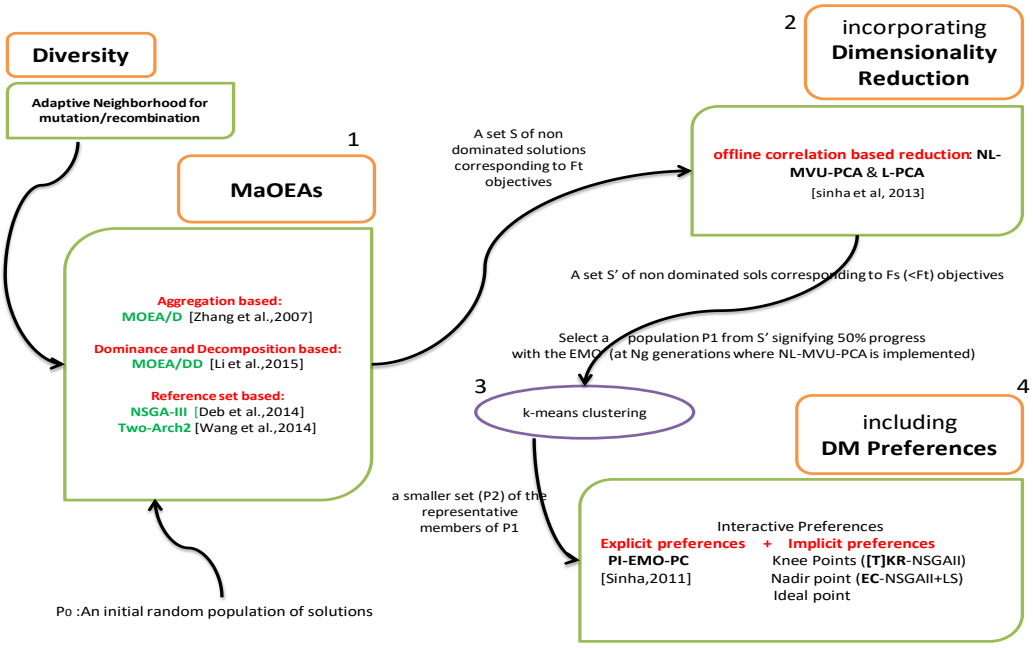


Figure B1 Les quatre étapes du schéma d'hybridation proposé

Dans ce schéma, quatre classes de MaOEA sont combinées : à base d'agrégation, à base de point de référence, à base de réduction et à base de préférence.

Au début, le MaOA est exécuté (MOEA/DD, NSGA-III ou Two-Arch2) tout en préservant la diversité en utilisant un mécanisme de voisinage adaptatif. Ainsi, un ensemble de solutions correspondant à l'optimisation de l'ensemble initial d'objectifs est obtenu. Sur cet ensemble, la réduction de dimensionnalité est effectuée et un ensemble de solutions correspondant à un plus petit ensemble d'objectifs est fourni. Ensuite, la procédure de préférence est établie.

En effet, les approches de préférence supposent qu'il n'existe pas d'objectifs redondants dans le problème donné (Saxena et al., 2013). Ainsi, dans notre approche, la procédure de réduction de la dimensionnalité est toujours effectuée avant d'appliquer les préférences du décideur (DM).

Étant donné que la procédure PI-EMO-PC (Sinha et al., 2014) nécessite une « fenêtre » de recherche suffisante, sa population d'entrée doit être choisie de manière à ce que la recherche converge vers une solution conforme aux préférences du DM. Pour choisir la population d'entrée pour notre procédure de préférence en garantissant un équilibre entre l'efficacité de calcul et la convergence vers une solution unique, nous suivons les mêmes étapes que celles

indiquées dans (Sinha et al., 2013) : La population d'entrée est une population intermédiaire prise après l'exécution de Ng générations (50% de progrès) après l'application de la réduction. Après avoir identifié cette population d'entrée pour la procédure de préférence, un ensemble plus petit représentant ses membres est considéré, en utilisant le clustering k-means. Ces membres sont les premiers membres de la population du PI-EMO-PC.

Notre algorithme de préférence hybride proposé PI-EMO-PC-INK: (PI-EMO-PC-Point idéal- point Nadir - points Knee)

Les préférences interactives sont les plus intéressantes puisqu'elles sont dynamiquement injectées dans le processus de sélection afin de guider en permanence la recherche. La figure B2 illustre la méthode hybride proposée basée sur la combinaison de préférences interactives implicites et explicites. Tout d'abord, le *point idéal* et le *point Nadir* sont respectivement obtenus en utilisant l'algorithme EC-NSGA-II (Deb et al., 2006) et en minimisant chaque objectif individuellement dans l'espace de recherche. Ensuite, un algorithme explicite basé sur PI-EMO-PC est exécuté si le DM a des préférences. Dans le cas contraire, un processus de préférence implicite visant à trouver des régions du genou basées sur le TKR-NSGA-II (Bechikh et al., 2011) est mis en œuvre. Lors de l'utilisation d'un algorithme explicite basé sur PI-EMO-PC, si le DM n'est pas satisfait mais qu'il est fatigué, la procédure exécute le processus pour trouver les régions *knee* précédemment indiquées. Par la suite, si le DM n'est pas encore fatigué, il modifie les niveaux d'aspiration du PI-EMO-PC afin d'intégrer de nouvelles informations sur ses préférences. Ces processus (PI-EMO-PC et TKR-NSGA-II) seront répétés si le DM n'est pas satisfait ou si le nombre maximal autorisé d'interventions autorisées n'est pas atteint. Dans les deux cas, lorsque le DM devient satisfait ou que le nombre maximal autorisé d'interventions autorisées est atteint, le processus global sera arrêté.

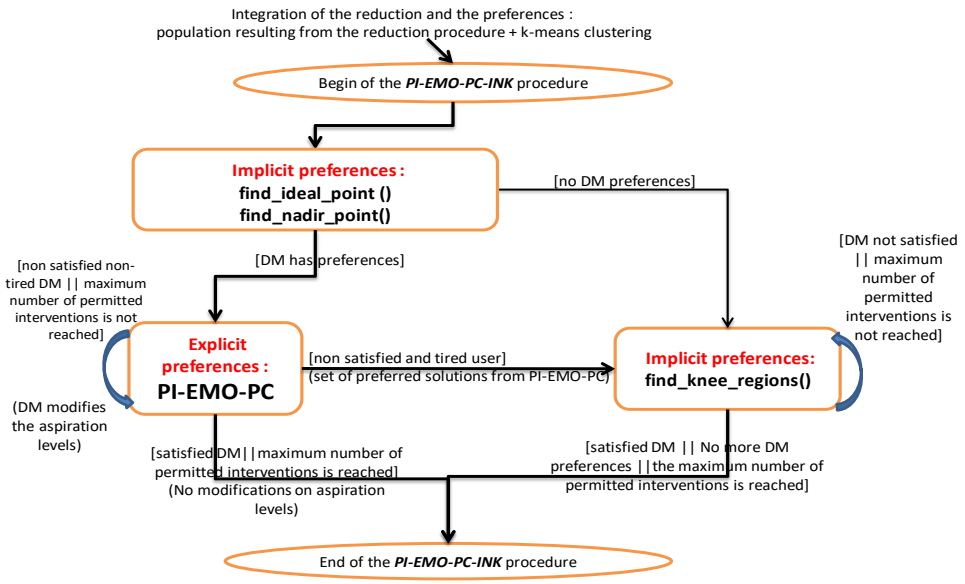


Figure B2 L'algorithme hybride proposé de préférences (PI-EMO-PC-INK)

PI-NSGA-III-VF (NSGA-III avec préférences interactives)

PI-EMO-VF (EMO progressivement interactive en utilisant la fonction de valeur) utilise une procédure générique qui peut être incorporée à tout algorithme d'optimisation multi-objectif évolutif (EMO). PI-EMO-VF utilise une fonction de valeur approximative qui est générée progressivement. Ainsi, après chaque nombre d'itérations (donc de générations) de l'algorithme EMO utilisé, un ensemble de solutions non dominées équitablement réparties est identifié, et on demande au DM de donner ses informations de préférences sur la relation entre les solutions. Dans un scénario idéal, le

DM fournit un classement complet des solutions (du meilleur au pire). Néanmoins, le DM peut donner une information partielle de préférence. Cette information aide à établir une fonction de valeur polynomiale croissante. Ensuite, suivant cette fonction de valeur, la condition d'arrêt est configurée.

Résultats de l'effet d'hybridation des EMOs avec une approche de réduction de dimensionnalité

A partir des résultats présentés dans le tableau B2, pour quatre et huit objectifs, les valeurs HV trouvées en utilisant l'approche de réduction de dimension sont plus élevées que celles obtenues sans utiliser cette méthode en raison de la réduction du nombre d'objectifs de huit à cinq dans notre problème du monde réel. De plus, le taux d'amélioration du MOEA/D dépasse clairement ceux des autres algorithmes.

Tableau B2 Meilleures, moyennes et pires valeurs du HV avant et après l'application de réduction de dimensionnalité

	Nbr d'Obj Initial	Nbr d'Obj après réduction	MOEA/D (PBI)	MOEA/DD	NSGA-III	Two_Arch2
Sans réduction	4 / 5	4 / 5	0.984426 0.976416 0.976328	0.983652 0.981426 0.976124	0.978863 0.978574 0.976231	0.976498 0.976422 0.976346
Avec L- PCA /NL-MVU-PCA		3	0.994975 0.991264 0.980023	0.982897 0.982542 0.982231	0.982896 0.982251 0.982033	0.989352 0.987144 0.986021
Sans réduction	8	8	0.969886 0.969815 0.969702	0.971841 0.969822 0.969723	0.966876 0.966525 0.966234	0.952886 0.952835 0.952803
Avec L- PCA /NL-MVU-PCA		4	0.971978 0.970951 0.970236	0.984986 0.984158 0.983943	0.969897 0.961152 0.960364	0.954237 0.953654 0.953028

Résultats de l'hybridation des EMO avec la réduction de dimensionnalité et les préférences

Dans cette section, nous examinons l'effet de l'application de notre approche proposée pour intégrer à la fois la réduction de la dimensionnalité et les préférences du DM. Dans nos tests, le HV est calculé avec une population égale à 1000. La fonction de scalarisation utilisée pour MOEA/D et MOEA/DD est PBI (0.5). La probabilité de mutation est de 1/400 avec un indice de 50, et la probabilité de recombinaison est de 0,9 avec un indice de 5. Les opérateurs adaptatifs de mutation et de recombinaison sont employés avec des parents voisins. 8 objectifs corrélés sont utilisés (et pour chaque expérience de N objectifs, au moins N/2 objectifs sont corrélés). Nous utilisons 250 points de référence pour NSGA-III. Après l'application de l'approche de réduction, la préférence est appliquée à l'ensemble réduit d'objectifs. Les Tableaux B3 et B4 montrent les spécifications des solutions finales. Chaque exécution a une population initiale différente, qui est le résultat de l'application de notre procédure de réduction sur l'EMO concernée. d_s est un paramètre défini par l'utilisateur représentant l'amélioration attendue des solutions obtenues à partir de la meilleure solution actuelle basée sur la fonction de valeur. TDMax est le nombre maximum d'appels de l'information de préférence introduite par le DM. Dans nos tests, $d_s = 0{,}01$ et TDMax = 30.

Tableau B3 Solutions moyenne obtenues (des valeurs objectives)

		Le point le plus préféré utilisé pour construire la valeur de la fonction objective	Valeurs moyenne des fonctions objectives f_i			
			MOEA/D	NSGA-III	Two_Arch2	MOEA/DD
f1	Nombre de noeuds nomades ajoutés	128.452	134.161	142.54	152.339	133.581
f2	Consommation d'énergie	3.857	3.998	4.021	4.056	3.962
f3 (redondant)	Coût de déploiement du matériel	85	88.468	96.184	93.923	88.646
f4 (redondant)	L'utilisation du réseau	1.00	0.946	0.796	0.849	0.962
f5	Taux de localisation	3.991	3.605	3.882	3.863	3.812
f6 (redondant)	Taux de couverture	5.865	4.189	4.984	4.235	5.572
f7	Durée de vie du réseau	4280	3885	3687	3956	4065
f8	Taux de connectivité	189.89	168.524	166.515	168.542	174.266

Tableau B4 Distance moyenne des solutions obtenues à partir des solutions les plus préférées

	MOEA/D	NSGA-III	Two_Arch2	MOEA/DD
Précision	0.234	0.419	0.468	0.023
Nombre d'évaluations de fonctions	6321	7945	8231	5895
Nombre d'appels requis du DM	TD_{Max}	26	25	16

Résultats des expérimentations sur PI-NSGA-III

Pour évaluer la connectivité (*f8* dans notre modélisation, voir la section 3.2.2 du rapport), la qualité des liens (*f10*) et la localisation (*f5*), la métrique RSSI est utilisée puisque le modèle de localisation utilisé repose sur le protocole RSSI et Distance-Vector Hop. Ainsi, plus la puissance RSSI reçue est élevé, meilleure est la localisation. La figure B3 illustre la moyenne valeur de RSSI (une valeur en [0, 256] convertible en dBm) entre les nœuds (RSSI entre chaque nœud i ∈ [1, 36] et les autres nœuds).

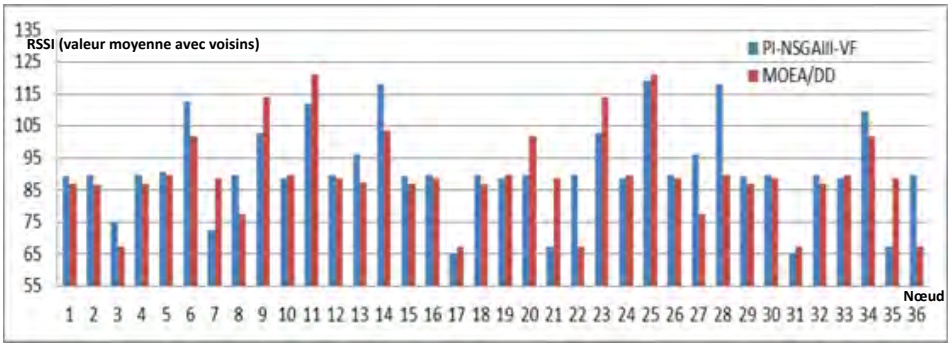


Figure B3 Taux moyens de RSSI

3.5 Hybridation entre l'algorithme de colonies de fourmis et l'algorithme génétique

Dans une quatrième approche, un nouvel algorithme hybride issu du comportement de recherche de fourmis et de la génétique est proposé. Il est basé sur la variante récente des algorithmes génétiques (NSGA-III) et l'algorithme de colonies de fourmis (ACO). C'est la première fois que le NSGA-III et l'ACO sont intégrés dans une plateforme hybride. Aussi, au contraire des hybridations classiques, ces deux algorithmes fonctionnent en même temps et interagissent en utilisant la même population (la population initiale du NSGA-III est la population construite par les fourmis dans la phase initiale de l'ACO). Ensuite, les étapes de l'algorithme des fourmis sont injectées dans le NSGA-III avec incorporation de plusieurs modifications du NSGA-III originale.

Malgré son efficacité, le NSGA-III a des difficultés à résoudre des problèmes d'optimisation mono-objectif et à deux objectifs. Ces difficultés concernent la faible pression de sélection que NSGA-III introduit dans les solutions non dominées d'une population lors de la résolution de problèmes à deux objectifs. De plus, les difficultés concernent la petite taille de la population et le processus de sélection aléatoire lors de la résolution de problèmes mono-objectifs (Ibrahim et al., 2016). De son côté, l'algorithme ACO présente un inconvénient principal qui concerne la convergence vers les optimum locaux (Sim et al., 2003).

L'idée est donc de proposer un schéma d'hybridation utilisant ces deux algorithmes (NSGA-III et ACO) pour tirer parti de leurs forces et remédier à leurs inconvénients. Lors de l'hybridation de ces deux algorithmes, la plupart des études (Huang and Chne, 2013), (Shen 2016) utilisent la version standard et basique de l'algorithme génétique. De plus, la plupart de ces études exécutent séquentiellement les deux algorithmes (le GA standard puis l'ACO, ou le contraire). Ainsi, la solution finale de l'un des deux algorithmes est la solution initiale de l'autre. Bien que ce schéma basique d'intégration améliore le taux de convergence ACO, celui-ci reste convergent de façon excessive, ce qui rend le problème de l'optimum local insoluble. Dans notre étude, nous proposons une plateforme où NSGA-III et ACO s'exécutent en même temps et interagissent avec la même population. Ainsi, les étapes de l'algorithme de la fourmi sont injectées dans le NSGA-III avec incorporation de plusieurs modifications de ce dernier. Parmi ces modifications, la population initiale du NSGA III devient la population construite par les fourmis dans la phase initiale de l'ACO. Il convient de mentionner que c'est la première fois que NSGA-III et ACO sont intégrés dans une plate-forme hybride. De plus, il s'agit de la

première plate-forme utilisant un algorithme génétique et ACO hybride pour résoudre le problème du déploiement 3D en intérieur des nœuds. L'algorithme proposé, appelé AcNSGA-III, est illustré dans Algorithme 3.6 de la section 3.3.5 du rapport. Il s'agit d'un algorithme fourmi-génétique hybride qui effectue une sélection ACO optimale des nœuds, une mise à jour dynamique des phéromones et une stratégie de mutation. Il accélère la convergence globale afin d'accélérer la recherche locale ce qui permet de trouver plus rapidement les solutions adaptées au problème de déploiement 3D. La capacité de recherche globale et le caractère aléatoire des opérateurs génétiques sont garantis, ce qui garantit la conduite de l'opération de l'opérateur génétique dans la génération de routes par les fourmis si l'ACO converge rapidement. Ceci permet à ce dernier de trouver les conditions de convergence. Enfin, puisqu'il y a une faible probabilité que les fourmis et le processus NSGA-III produisent le même individu dans la même itération, l'individu est ajouté à la population à moins qu'il n'y existe pas.

Résultats : afin de mesurer la localisation, la métrique RSSI est utilisée puisque notre modèle de localisation est basé sur l'hybridation entre le RSSI et le protocole de localisation DVHop. Ainsi, plus le RSSI est élevé, meilleure est la localisation. Un voisin peut être ajouté à la table de voisinage d'un nœud uniquement si la valeur RSSI du nœud détecté est supérieure à un seuil prédéfini. Selon nos expérimentations réelles, on a fixé cette valeur à 90 (valeur RSSI entre 0 et 256 convertible en dBm). Initialement, les niveaux RSSI sont basés sur nos expériences empiriques. Ensuite, afin de garantir le dynamisme au sein du réseau, une perturbation de la valeur de RSSI est introduite via une fonction aléatoire. La figure B4 montre, pour différents nombres d'objectifs pris en compte par les algorithmes testés, la moyenne des valeurs RSSI mesurées pour tous les nœuds en relation avec le nœud mobile.

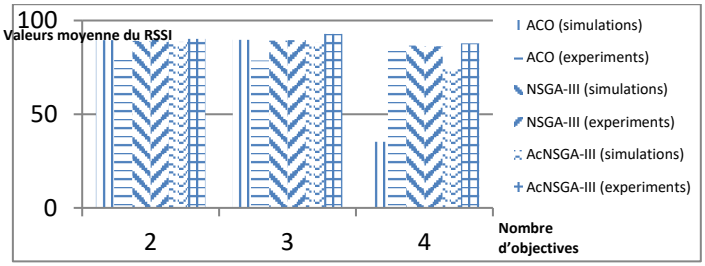


Figure B4 Comparaison des taux moyens de RSSI

3.6 Nouveau concept d'accent d'oiseaux et hybridation entre l'essaim de particules à base d'accent d'oiseaux et les SMA

Dans une cinquième approche, l'algorithme d'optimisation de l'essaim de particules basé sur un nouveau concept d'accent d'oiseaux (appelé acMaPSO) est proposé. Le nouveau concept d'accent d'oiseaux est basé sur un ensemble d'oiseaux qui sont séparés en différents groupes d'accentuation par leur habitation régionale et qui sont classés en groupes selon leurs manières de chanter. Pour conserver la diversité de la population pendant la recherche, ce nouveau concept d'accent d'oiseau est introduit pour évaluer la capacité de recherche des particules dans leurs zones locales. Pour assurer que la recherche s'échappe des optimums locaux, les particules les plus expertes (les parents) "meurent" et sont régulièrement remplacées par des nouvelles particules qui sont aléatoirement générées.

On propose également de tester cette hybridation dans un milieu distribué. Pour cela, on propose d'hybrider l'algorithme acMaPSO avec un système multi-agent (SMA). La nouvelle variante (nommée acMaMaPSO) tire parti de la distribution et de l'interactivité des agents. L'architecture multi-agents est décentralisée puisqu'il s'agit d'un réseau ad-hoc. Cette architecture renferme trois types d'agents : un agent environnement, des agents d'essaim et des

agents oiseaux (ou agents particules). Ces agents possèdent des connaissances, des buts, des capacités et des plans d'actions différentes.

L'algorithme acMaPSO proposée: incluant le concept d'accent d'oiseaux sur le MaOPSO. Les modifications proposées de l'OEP multi-objectif standard visent principalement à surmonter les difficultés rencontrées lors de la résolution de problèmes complexes du monde réel ayant généralement plusieurs optimums locaux. Dans notre approche, des changements sont faits dans la topologie de l'essaim : afin d'éviter le caractère prématuré de l'algorithme OEP standard (Kennedy and Eberhart, 1995), en plus de l'utilisation de la meilleure position globale (gbest) et de la meilleure position personnelle de la particule (pbest), nous utilisons la meilleure position de la zone locale autour de la particule, appelée le meilleur cluster (cbest). En effet, selon des recherches récentes en biologie (Oesel et al., 2017), les oiseaux chanteurs ont des accents régionaux exactement comme les êtres humains. En effet, la capacité de chanter et de créer une chanson complète que les oiseaux possèdent est héritée en grande partie de leurs parents. Des expériences ont montré que si ces oiseaux sont élevés en silence, ils ne peuvent que crier. De plus, les oiseaux de différentes régions développent des accents distincts. En imitant ce concept biologique, nous proposons une OEP basée sur une topologie des catégories d'accent d'oiseaux chanteurs (acMaPSO). L'idée est que chaque groupe d'accent possède des paramètres différents pour accélérer la convergence, ce qui améliore la prévention des optimums locaux. Pour évaluer les capacités de recherche des particules dans leurs zones locales, cet algorithme est basé sur ce nouveau concept d'accent où les particules appartiennent à différentes communautés ou groupes. Pendant la recherche, les particules sont séparées en groupes en fonction de leurs accents, comme le montre la figure B5. Pour conserver la diversité de la population, les particules de chaque catégorie d'accent peuvent sélectionner comme voisins seulement les particules les moins expérimentées (de leur propre groupe ou des autres groupes).

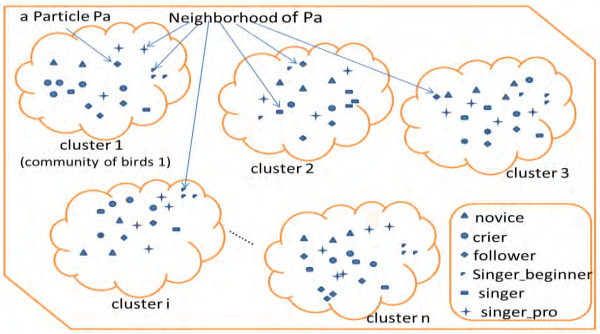


Figure B5 Le choix du voisinage d'une particule Pa

acMaMaPSO : Un algorithme OEP hybride, multi-objectif, multi-agent, basé sur l'accent d'oiseaux

acMaMaPSO est un nouvel algorithme qui combine les principales caractéristiques du acMaPSO et SMA. En effet, l'hybridation du PSO avec les SMA allie l'autonomie et les capacités d'apprentissage du SMA à la simplicité de l'OEP. En conséquence, les particules deviennent plus autonomes, plus intelligentes et plus capables de tirer profit de leur environnement. En collaborant entre eux, les agents développent une société pour atteindre un but commun ainsi que leurs propres objectifs individuels. Le processus de prise de décision du groupe dans le SMA correspond à la nature fondamentale d'une particule dans l'OEP. Par conséquent, l'hybridation proposée offre la possibilité de calculer et d'optimiser des problèmes complexes

L'architecture SMA proposée : l'architecture SMA proposée repose sur trois types d'agents : l'agent d'environnement (*agEnv*), l'agent d'oiseau (*agBird* ou *agParticle*) et l'agent swarm (*agSwarm* ou *agCommunity*). En effet, après l'initialisation des paramètres, *agEnv* affecte

pour chaque *agBird* une position de départ dans l'espace problème et un ensemble de voisins. Ensuite, les agents d'oiseaux commencent à rechercher une solution optimale jusqu'à ce qu'un nombre maximum d'itérations soit atteint ou qu'une bonne fonction *fitness* soit trouvée. Chaque *agBird* vérifie si sa position actuelle est déjà visitée. Si oui, il n'évalue pas sa fonction de fitness et interroge ses voisins sur leurs *pBest* et leurs emplacements, puis interroge l'*agEnv* sur les emplacements des clusters voisins afin de calculer son *pBest* actuel et de mettre à jour sa position et sa vitesse. Si la position actuelle n'est pas visitée, l'oiseau (*agBird*) évalue la fonction *fitness*, marque le point actuel comme visité et met à jour sa position et sa vitesse. La procédure globale est illustrée à la figure B6.

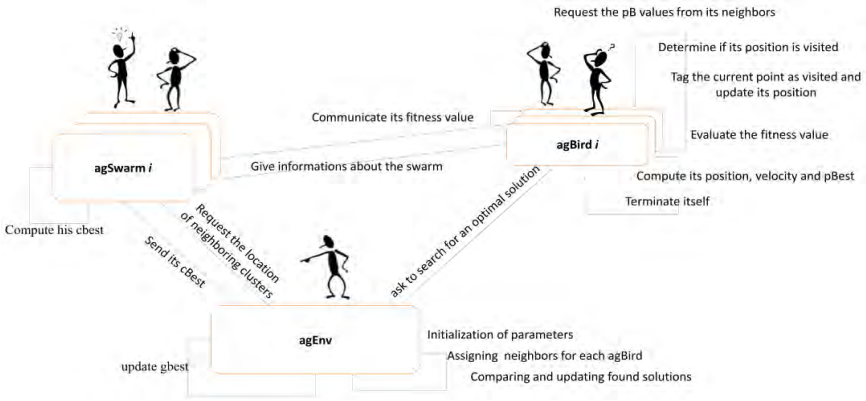


Figure B6 L'architecture SMA proposée

Pour évaluer le coût de déploiement (*f1* dans notre modélisation, voir section 3.2.2) et la localisation (*f5*), la métrique RSSI est utilisée. La figure B7 illustre, pour différents nombres d'objectifs (à satisfaire par les algorithmes testés), les valeurs moyennes du RSSI mesurés pour tous les nœuds en connexion avec le nœud mobile. Ce RSSI moyen est une valeur dans [0, 256] convertible en dBm.

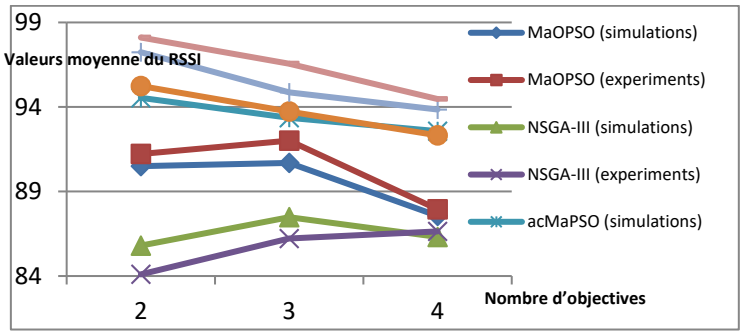


Figure B7 Taux moyens de RSSI des nœuds en connexion avec le nœud mobile

Pour mesurer la couverture (*f6*), le FER (Frame Error Rate) est utilisé comme une métrique pour évaluer la qualité des liens entre les nœuds. Ainsi, plus le FER est bas, meilleure est la couverture. La figure B8 illustre, pour un nombre différent d'objectifs à satisfaire par les algorithmes testés, la moyenne des taux FER mesurés pour tous les nœuds en connexion avec (détectant / détecté par) le nœud mobile.

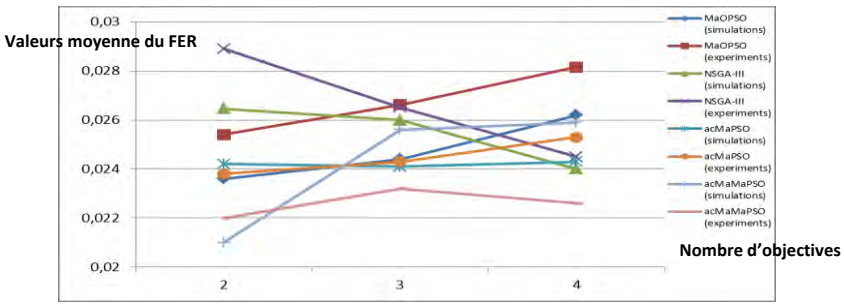


Figure B8 Taux moyens de FER des nœuds en connexion avec le nœud mobile

4 Conclusions et interprétations

Les principales contributions de nos travaux de recherche et leurs résultats peuvent être résumées comme suit :

- Une étude bibliographique approfondie portant sur les méthodes d'optimisation du déploiement dans les réseaux de capteurs sans fil, particulièrement les algorithmes qui prennent en considération le déploiement 3D à l'intérieur. Cette étude couvre les méta-heuristiques à base d'essaim (optimisation par Essaim Particulaire, optimisation par colonies de fourmis), les algorithmes génétiques, la recherche tabou, le recuit simulé. L'étude traite aussi le cas mono-objectif et multi-objectif ; le cas statique et dynamique du déploiement.
- Une formulation mathématique qui vise la modélisation de la problématique. Il s'agit de fixer la fonction objective à optimiser, les variables de décision et les différentes contraintes à prendre en considération. Notre objectif est de minimiser le nombre de nœuds capteurs à utiliser et la consommation de l'énergie. Simultanément, on vise ainsi la maximisation de la durée de vie du réseau, mais aussi la couverture, la localisation et la connectivité.
- On propose aussi une mutation adaptative et des opérateurs de recombinaison qui favorisent le croisement avec le voisinage, et l'intégration d'un concept de fonctions scalarisation multiples dans les EMO à base de Pareto pour lutter contre l'inefficacité de la recombinaison et à l'augmentation des coûts en temps et en espace dans le cas des problèmes réelles à plusieurs objectifs. Les résultats obtenus montrent que pour la plupart des cas, l'algorithme d'agrégation (MOEA/DD) est plus efficace que les autres algorithmes testés (∈-NSGA-II and NSGA-III) pour résoudre du déploiement 3D à l'intérieur. De plus, les résultats prouvent que la méthode adaptative de sélection des opérateurs de recombinaison et de mutation avec des restrictions de voisinage améliore la performance des algorithmes.
- Une première hybridation justifiée qui combine deux procédures basées sur les préférences des utilisateurs (PI-EMO-PC) et la réduction de dimensionnalité (MVU-PCA) avec des algorithmes d'optimisation appartenant à trois classes : algorithmes à base d'agrégation (Two_Arch2), à base de points de référence (MOEA/DD, NSGA-III) et à base de décomposition (MOEA/D). Pour évaluer les nouveaux algorithmes hybrides proposés, on a utilisé la métrique HV sur notre problème de déploiement ; et la métrique IGD sur des problèmes de tests théoriques tels que DTLZ. Ensuite, pour valider les observations théoriques, un ensemble de prototypages a été réalisés. Les résultats montrent la performance supérieure des algorithmes hybrides modifiés par rapport aux algorithmes originaux. On a constaté aussi d'autres découvertes intéressantes telles que la performance supérieure du MOEA/D par rapport à NSGA-III dans le cas d'objectifs corrélés.
- Une deuxième hybridation est proposée, qui combine une approche interactive d'incorporation des préférences de l'utilisateur (PI-EMO-VF) avec l'algorithme NSGA-III. Les résultats ont montré que l'algorithme hybride proposé dépasse pour certaines métriques d'évaluation (comme le nombre de voisins), la performance d'un autre algorithme récent d'optimisation avec plusieurs objectifs (MOEA/DD) sachant que la littérature (voir (Li et al. 2015) et (Yuan et al., 2016)) affirme que MOEA/DD donne de meilleurs résultats que le NSGA-III standards sur les problèmes théoriques.

- Une troisième hybridation est proposée, qui combine, pour la première fois, l'algorithme d'optimisation de colonies des fourmis (ACO) avec une variante récente des algorithmes génétiques (NSGA-III). L'avantage de ce schéma d'hybridation est que, au contraire des hybridations classiques, les deux algorithmes hybridés s'exécutent en même temps et peuvent interagir avec la même population. A partir des résultats, on peut déduire que l'algorithme proposé (acNSGA-III) surpasse les algorithmes NSGA-III et le ACO standards.
- Une autre modification proposée, consiste à introduire un nouveau concept d'accent d'oiseau dans l'algorithme d'optimisation de l'essaim de particules. Le nouvel algorithme est nommé acMaPSO. Ce nouveau concept se base sur l'idée de séparer l'essaim (la communauté des oiseaux) en différents groupes selon leurs accents. En effet, pour conserver la diversité de la population durant le processus de recherche de l'algorithme, ce concept d'accent évalue la capacité de recherche locale des particules et classifie les particules en différents groupes d'accent selon leur manière commune de chanter. La sélection des voisins de chaque particule peut s'effectuer parmi les particules du même groupe ou parmi d'autres groupes. Pour s'échapper aux optima locaux, les particules âgées "meurent" et sont remplacées périodiquement par des nouveau particules générées aléatoirement.
- Une quatrième hybridation proposée, consiste à distribuer l'algorithme acMaPSO en l'hybridant avec un multi-agent system (SMA). Le nouveau algorithme (acMaMaPSO) bénéficie de l'interactivité des particules (devenues des agents) et de la distribution. L'architecture multi-agents proposée se base sur trois types d'agents qui possèdent des connaissances et des plans d'actions différents : un agent environnement, des agents oiseaux (ou agents particules) et des agents d'essaim. Les deux algorithmes acMaPSO et acMaMaPSO sont très compétitifs par rapport à l'algorithme MOPSO. Les résultats obtenues montrent que les deux algorithmes proposés (acMaPSO et acMaMaPSO) atteignent (et dépassent pour des métriques d'évaluation spécifiques telles que le nombre de voisins), la performance de l'algorithme PSO standard et l'algorithme NSGA-III.

Les schémas d'hybridation précédemment mentionnés sont tous validés par des résultats numériques qui ont permis l'évaluation des algorithmes avec des métriques telles que l'hypervolume. Ensuite, des simulations complétées par des expérimentations réelles sur des *testbeds* ont été proposées. Enfin, les simulations sont confrontées aux expérimentations pour évaluer le comportement des algorithmes et prouver leur stabilité et leur efficacité.

Pour résumer, les indicateurs de performances utilisés pour justifier notre travail, les simulations et les prototypages réels mettent en évidence le très bon comportement de nos approches proposées et montrent l'efficacité de l'hybridation.

5 Perspectives et directions de recherche futures

Les perspectives de nos travaux sont multiples et plusieurs questions ouvertes restent à explorer. Nos perspectives concernent principalement deux classes : La première s'intéresse au développement de nouveaux schémas d'hybridation entre EMO et autres approches de résolution de déploiement. La seconde concerne les défis en relation avec le réseau, les expérimentations réelles et le prototypage.

Perspectives d'hybridation :

- Selon ces auteurs (Deb et al., 2010), la procédure PI-EMO-VF renferme un nombre assez élevé de paramètres à initialiser auparavant par l'utilisateur, ce qui le rend sensible au paramétrage initial. D'où l'importance de proposer une version du PI-EMO-VF avec moins de paramètres.
- On peut aussi intégrer d'autres MOEA récents et d'autres procédures de réduction (autre que celle de MVU-PCA utilisé déjà) dans notre plate-forme pour évaluer leurs performances. A

cet égard, on peut incorporer l'algorithme KnEA (Zhang et al., 2015) et la procédure de réduction FS (Feature Selection) (Mitra et al., 2002).
- Bien que notre plateforme d'incorporation de réduction et préférences puisse réduire le nombre d'objectifs et par la suite la complexité du problème, elle semble elle-même complexe à implémenter et à mettre en œuvre. Par conséquent, pour montrer l'apport de son utilisation, une étude de la complexité algorithmique de notre approche hybride peut être menée.
- L'intégration des préférences dans des méta-heuristiques autre que les EA s'avère une piste prometteuse car différentes méta-heuristiques basées sur la population ont prouvé leurs performances dans la résolution de MOPs. On peut citer à ce propos le PSO et l'AIS.
- Afin d'aider le DM à naviguer dans des espaces multidimensionnels des problèmes à plusieurs objectives, des recherches peuvent être menées en ce qui concerne le développement de meilleurs outils de visualisation.
- Dans la suite du travail présenté dans cette thèse, on poursuit actuellement nos travaux en proposant un nouvel algorithme d'optimisation basé sur l'imitation des comportements des animaux sauvages dans leurs territoires, et son application sur notre problématique.

Perspectives applicatives de réseau et de prototypage :
- Un des avantages du banc d'essai Ophelia que nous avons utilisé est qu'il est plus réaliste en termes de contraintes et conditions d'utilisation réelles, qu'une plate-forme ayant un grand nombre de nœuds uniformes tels que SmartSantander (Sanchez et al., 2014), IoTLab (Fleury et al., 2015) ou INDRIYA (Doddavenkatappa et al., 2011). Ces plateformes nous permettent de tester nos algorithmes en utilisant un nombre de nœuds plus grand (jusqu'à 1024 nœuds). Dans de futures études, afin de comparer nos résultats d'Ophelia avec d'autres plateformes de prototypages et afin de prouver l'évolutivité de notre approche, des tests sur la plateforme IoTLab sont envisageables puisque cette dernière permet de tester les mêmes métriques de nos expériences (Qualité de lien, RSSI,... etc)
- D'autres pistes de recherche peuvent être investiguées comme la résolution du redéploiement dynamique des nœuds avec prise en considération d'autres objectifs. Nous visons aussi à intensifier le réseau déployé en ajoutant de nouveaux nœuds pour mieux satisfaire la contrainte de localisation qui exige quatre voisins pour chaque cible. On pourrait refaire par la suite nos expériences pour avoir une idée sur l'influence de la densité du réseau sur les résultats. Néanmoins, l'implémentation de la localisation par DV-Hop RSSI est aussi intéressante à évaluée réellement.
- Une autre contribution future concerne l'implémentation d'un modèle énergétique plus réaliste avec OMNeT++, en se basant sur la gestion des valeurs BO et SO (Farhad et al., 2016) du protocole CSMA/CA 802.15.4 par exemple.
- Malgré son efficacité, OpenWiNo a le défaut de manque de bibliothèques qui implémentent les protocoles standards, d'où l'intérêt d'inclure et implémenter d'autres protocoles de transmission et technologies dans OpenWiNo.

Finalement, vu le succès de l'utilisation de l'optimisation multi-objectif dans la résolution de plusieurs problèmes du monde réel ayant des objectifs souvent nombreux et conflictuels, le moment est venu pour pousser son utilisation dans des domaines d'applications différents, dynamiques et complexes.

Contributions to the optimized deployment of connected sensors on the Internet of Things collection networks

IoT collection networks raise many optimization problems; in particular because the sensors have limited capacity in energy, processing and memory. In order to improve the performance of the network, we are interested in a contribution related to the optimization of the 3D indoor deployment of nodes using multi-objective mathematics models relying on hybrid meta-heuristics. Therefore, our main objective is to propose hybridizations and modifications of the optimization algorithms to achieve the appropriate 3D positioning of the nodes in the wireless sensor networks with satisfaction of a set of constraints and objectives that are often antagonistic. We propose to focus our contribution on meta-heuristics hybridized and combined with procedures to reduce dimensionality and to incorporate user preferences. These hybridization schemes are all validated by numerical tests. Then, we proposed simulations that are completed by, and confronted with experiments on real testbeds.

Keywords: 3D indoor deployment, ant colony algorithm, dimensionality reduction, experimental validation, genetic algorithms, hybridization, IoT collection networks, meta-heuristics, optimization, particle swarm optimization, user preferences.

Contributions au déploiement optimisé des capteurs connectés dans les réseaux de collecte de l'Internet des Objets

Les Réseaux de collecte de l'IoT soulèvent de nombreux problèmes d'optimisation, à cause des capacités limitées des capteurs en énergie, en traitement et en mémoire. Dans l'optique d'améliorer la performance du réseau, nous nous intéressons à une contribution liée à l'optimisation du déploiement 3D d'intérieur des nœuds sur les réseaux de capteurs sans fil en utilisant des méta-heuristiques hybrides se basant sur des modèles mathématiques multi-objectif. L'objectif principal est donc de proposer des hybridations et modifications des algorithmes d'optimisation dans le but de réaliser le positionnement 3D adéquat des nœuds dans les réseaux de capteurs sans fil avec satisfaction d'un ensemble de contraintes et objectifs qui sont souvent antagonistes. Nous proposons d'axer notre contribution sur les méta-heuristiques hybrides et combinés avec des procédures de réduction de dimentionalité et d'incorporation de préférences des utilisateurs. Ces schémas d'hybridation sont tous validés par des résultats numériques de test. Ensuite, des simulations complétées par; et confrontées à ; des expérimentations sur des testbeds réelles.

Mots-clés: algorithmes génétiques, algorithme de colonie de fourmis, déploiement 3D à l'intérieur, hybridation, méta-heuristique, optimisation par essaim de particules, préférences d'utilisateur, réduction de la dimensionnalité, réseaux de collecte IoT, validation expérimentale.