



HAL
open science

A new multi-agent particle swarm algorithm based on birds accents for the 3D indoor deployment problem

Sami Mnasri, Nejah Nasri, Adrien van den Bossche, Thierry Val

► To cite this version:

Sami Mnasri, Nejah Nasri, Adrien van den Bossche, Thierry Val. A new multi-agent particle swarm algorithm based on birds accents for the 3D indoor deployment problem . ISA Transactions, 2019, 91, pp.262-280. 10.1016/j.isatra.2019.01.026 . hal-02571791

HAL Id: hal-02571791

<https://hal.science/hal-02571791>

Submitted on 13 May 2020

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

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



Open Archive Toulouse Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in: <http://oatao.univ-toulouse.fr/24756>

Official URL: <https://doi.org/10.1016/j.isatra.2019.01.026>

To cite this version: Mnasri, Sami and Nasri, Nejah and Van Den Bossche, Adrien and Val, Thierry *A new multi-agent particle swarm algorithm based on birds accents for the 3D indoor deployment problem.* (2019) ISA Transactions, 91. 262-280. ISSN 0019-0578

Any correspondence concerning this service should be sent to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

Practice article

A new multi-agent particle swarm algorithm based on birds accents for the 3D indoor deployment problem[☆]

Sami Mnasri^{a,*}, Nejah Nasri^b, Adrien van den Bossche^a, Thierry Val^a

^a UT2J, CNRS-IRIT (RMES), University of Toulouse, Toulouse, France

^b ENIS, LETI, University of Sfax, Sfax, Tunisia

H I G H L I G H T S

- Proving the efficiency of optimization algorithms in solving real-world problems.
- A new concept of accent birds introduced to the particle swarm optimization.
- A new hybrid scheme that integrates PSO and MAS.
- The hybridization improves the performance of the original tested algorithms.
- A comparison between simulation and experimental validation is given.

A R T I C L E I N F O

Keywords:

Accent based PSO
Multi-agent
Many-objective optimization
Experimental validation
3D indoor deployment
DL-IoT collection networks

A B S T R A C T

The 3D indoor deployment of sensor nodes is a complex real world problem, proven to be NP-hard and difficult to resolve using classical methods. In this context, we propose a hybrid approach relying on a novel bird's accent-based many objective particle swarm optimization algorithm (named acMaPSO) to resolve the problem of 3D indoor deployment on the Internet of Things collection networks. The new concept of bird's accent is presented to assess the search ability of particles in their local areas. To conserve the diversity of the population during searching, particles are separated into different accent groups by their regional habitation and are classified into different categories of birds/particles in each cluster according to their common manner of singing. A particle in an accent-group can select other particles as its neighbors from its group or from other groups (which sing differently) if the selected particles have the same expertise in singing or are less experienced compared to this particle. To allow the search escaping from local optima, the most expert particles (parents) "die" and are regularly replaced by a novice (newborn) randomly generated ones. Moreover, the hybridization of the proposed acMaPSO algorithm with multi-agent systems is suggested. The new variant (named acMaMaPSO) takes advantage of the distribution and interactivity of particle agents. Experimental, numerical and statistical found results show the effectiveness of the two proposed variants compared to different other recent state-of-the-art of many-objective evolutionary algorithms.

1. Introduction

The deployment of nodes is the first phase in the design and implementation of a WSN (Wireless Sensor Network). Thus, it significantly influences its performance. Indeed, the deployment is a strategy which aims to determine the number of nodes, their

positions and the topology of the network. In this study, we are interested in the deployment of nodes in a three-dimensional space that reflects the real topology of the RoI (region of interest) better than the two-dimensional case. We are particularly interested in improving the initial 3D indoor deployment by adding new nodes while optimizing a set of objectives such as coverage, connectivity, localization, quality of links and network utilization. These objectives will be detailed in the modeling section. To resolve the deployment problem in WSN, the topology of the network can be modeled as an identification problem in a graph as addressed in [1] and [2]. In this static case, the deployment algorithm is run off-line with no motion during the evolution of the algorithm and the positions generated when the process converges represent the final topology. Some other on-line deployment strategies aim to

[☆] This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

* Correspondence to: bureau C107, Batiment C, 1 Place Georges Brassens, 31700 Blagnac, Toulouse, France.

E-mail addresses: Sami.Mnasri@fsgf.rnu.tn (S. Mnasri), nejah.nasri@isecs.rnu.tn (N. Nasri), vandenbo@irit.fr (A.V.D. Bossche), val@irit.fr (T. Val).

move the nodes while the algorithm is running, to improve some criteria of the deployment like the coverage. Such methods can be applied to mobile WSN. In this paper, we are interested in a dynamic context where the mobility of nodes is used to evaluate the performance of the network by collecting the values of some metrics (such as the RSSI (Received Signal Strength Indication), the FER (Frame Error Rate), the number of neighbors or the energy consumption rate) from the network in different locations. These locations are taken by the mobile node to evaluate the effect of the positions of the nomad nodes on the performance of the network. Nomad nodes are added following the proposal of the tested optimization algorithms. Although, the issue in our paper is offline since there is no programming on sensors and EMO (evolutionary multi-objective optimization) algorithms give solutions offline. Indeed, we can compare the optimization algorithms (between them) according to their proposed deployment solutions. In our simulations, the mobile node is used to trigger the simulation process (by sending a first message, then, as explained in the experimental/simulation scenario (Section 5.2.3), the used AODV (Ad hoc On-demand Distance Vector) routing protocol create a flow of messages in the network, which allows us to evaluate these performances after adding the nomad nodes). In our experiments, one or more mobile node(s) move(s) in the RoI to assess the network performance after adding the nomad nodes. The topology can be considered as dynamic in the sense that we have mobile nodes and a set of nomad nodes to add after the initial deployment.

Actually, DL-IoT (Device Layer-Internet of Things) collection networks represent the mutation of the WSNs to IoT (Internet of things). DL-IoT is a network of connected objects (called 'nodes' in WSNs) which is used to collect information. Therefore, our problem is to deploy a 3D indoor DL-IoT. It is a scenario in which a set of autonomous and connected entities (robots or peripherals) interact and each one has a unique identifier. These objects communicate with each other using protocols such as 802.15.4 or Bluetooth. Indeed, the WSN ensures the hardware communication and the transmission of the real values detected by the sensors, while the IoT provides the manipulation of this data and the decision-making. Our modeling and approach can be applied in the context of WSNs and that of IoT.

To evaluate the efficiency and the performance of the new solutions, protocols and algorithms, it is recommended to use real-world platforms, often known as "testbeds", to test protocols with a large number of real nodes. Indeed, the choice of using real experiments rather than simulations is justified by the access to the hardware and the prototyping of the communication devices which are more and more simple. It is also justified by the realism of the obtained results. In this context, the deployment of connected objects in prototypes and the evaluation of their performance with classical network metrics and human feeling is a very interesting approach to researchers. Among real-world platforms, SensLab which is currently known as FIT/IoT-LAB [3] and SmartSantander [4] which is a platform for smart cities. Each of these testbeds has several thousand available nodes. Moreover, INDRIYA [5] and TWIST [6] allowing the deployment of nearly 200 nodes with several levels. Most of the mentioned testbeds use a physical layer based on protocols that are standardized by IEEE 802.15.4-2006, with 868 MHz or 2.4 GHz. Thus, in contrast with different works based on formal calculations, theoretical hypotheses or simulations, the aim of this study is to characterize as finely as possible the real world using real physical nodes (36 Teensy nodes in our prototyping platform that will be presented in the experiment section).

To find the best positions of the connected objects/nodes, while satisfying the above-mentioned objectives, our approach introduces a modified PSO (particle swarm optimization) algorithm

which is based on a new concept of accent birds. Moreover, hybridization between MAS (multi-agent systems) and PSO is proposed.

The motivation of this study using this approach raised from the lack of studies resolving many-objective deployment problems using meta-heuristic optimization approaches although the efficiency of these latter compared to other classical deployment algorithms for the many-objective case [7].

The main proposed contributions in this paper can be summarized as follows:

- A mathematical formulation is introduced to model the complex real-world problem (the 3D indoor deployment) with many conflict objectives to satisfy.

- The proposition of acMaPSO (accent-based many-objective PSO) which is a modification of the MaOPSO (many-objective PSO) algorithm that introduces a new concept in the PSO (the accent of birds) and combines a local search mechanism with a global one to help escaping from local optima. Moreover, a specific particle experience concept in PSO is proposed to assess the experience of each particle in the swarm. Then, based on this new measure, an accent-group topology structure is designed to manage the swarm diversity.

- The proposition of acMaMaPSO (accent-based many-objective multi-agent PSO) which introduces hybridization between PSO and MAS that benefits from the advantages of the two approaches. The hybridization of a multi-objective PSO with MAS exists (see Section 2.2), but there is no many-objective version of the PSO which is hybridized with MAS.

- There is no implementation of this hybridization on a real-world application; we propose as an application the 3D indoor deployment in WSN with an experimental validation on real testbeds. The performances of the proposed algorithms (acMaPSO and acMaMaPSO) are compared to those of the MaOPSO and the NSGA-III, followed by a comparison between the simulation results and the real prototyping experiments. The interpretation of the obtained results is also provided.

Other innovative ideas of this paper concerns mainly:

- Proving the efficiency of optimization algorithms in solving complex real world problems.

- A comparison between simulation and experimental validation is given.

- The proposed hybridization improves the performance of the original tested algorithms.

- The mathematical formulation of the problem relies on real hypotheses and assumptions.

The remainder of the paper is organized as follows: A set of related works are discussed in Section 2 to prove the suitability of our method for the deployment problem. The modeling of the problem is given in Section 3. The novel accent concept and the hybrid scheme of the PSO with MAS are presented in Section 4. The numerical results of the proposed algorithms and a set of experiments on real testbeds which are compared to simulations and followed by statistical and execution-time tests are detailed in Section 5. To sum up, a conclusion and different perspectives are discussed in Section 6.

2. Related works

2.1. Related works on the 2D-3D deployment problem using optimization algorithms

Several recent works proposed scalable optimization algorithms for efficient deployment of WSN nodes. In [8], Banimelhem et al. propose a genetic algorithm (GA) to solve the problem of coverage holes and deterministic 2D deployment in WSNs. The goal is to minimize the number of mobile nodes. However, no

mathematical modeling is given to the problem. In [9], authors suggest a multiobjective approach relying on Pareto dominance genetic algorithms to resolve the problem of designing butterfly valve. However, the proposed approach is based on a simple standard genetic algorithm. Besides, simulations are carried out without using known evaluation metrics to assess the performance of the approach. In [10], Danping et al. propose a multi-objective evolutionary algorithm combined with a low-cost heuristic to solve the problem of signal and radio propagation in indoor with a 3D deployment scenario. The objective is to optimize the coverage, the link quality, the lifetime and the cost of hardware. However, the scalability of the proposed approach is not proved. In [11], Ko et al. suggest a GA relying on a parsing crossover scheme to solve the deployment in 3D irregular terrains. The objective is to maximize the probabilistic point coverage and the global coverage. Although, they did not prove that the proposed parsing crossover strategy is better than the original genetic approach. Authors in [12] propose an algorithm which is based on a harmony search to optimize the number of deployed sensors and the coverage. The disadvantage of this work is that the proposed network model is simplistic. Also, only two objectives are considered and the validation of the approach is based only on Matlab tests, without a scenario of simulation or real-world experiments. In [13], the problem of 3D positioning of nodes is studied while optimizing the number of nodes, the consumption of energy, the coverage area and the localization rate. The proposed approach is based on two evolutionary algorithms (MOEA/D and NSGA-III). However, this study is based on a simple comparison between the two mentioned algorithms and the evaluations only rely on simulations without experimental validations.

2.2. Related works on the hybridization of PSO and MAS

Different other studies suggest various schemes to hybridize PSO and MAS. Kolomvatsos and Hadjieftymiades propose in [14] a PSO algorithm relying on intelligent and simultaneous negotiations between a buyer agent and seller ones to achieve a purchase transaction. The PSO algorithm was performed on each thread with a swarm intelligence strategy to get the optimal agreement. However, only two objectives are considered in this study. In [15], a MAS based on a particle swarm optimization approach (MAPSO) is used to solve the problem of distribution of the economic charge of power system. Indeed, each agent cooperates with its neighbors to adjust its global search capability and its local exploration capability. Although the results indicate that this algorithm has a high accuracy and speed of convergence compared to other evolutionary algorithms, this algorithm must be tested on a real problem with a larger number of objectives and dimension of the problem. Moreover, the modeling of the problem is not presented. In [16], the authors propose another agent-based PSO algorithm. With a higher degree of learning and an asynchronous execution, the particles in this algorithm have more autonomy. The environment of these agent particles is modeled as a cluster of non-optimal points. The authors also implemented a parallel variant of the proposed PSO algorithm which is more efficient but more complicated. Another study that combines the PSO with MAS and GAs was proposed in [17]. The authors suggest the integration of the mentioned approaches to solve the problem of automatic generation of test paper in a parallel computing environment. The proposed multi-agent architecture is based on a remote node, a central one and a set of agents called *TPAgents* that control the evolution operations of each population generation in the GA. To minimize the cost of communication between the nodes, a fitness evaluation is performed by *TPAgents* at the level of local nodes. Only the best local particle and its corresponding local fitness are sent to the central node to calculate the best global particle.

To escape from local optima, the best global particle is randomly transmitted to the remote node. Despite the fact that the proposed approach is efficient, the scalability of this system is not proven and the number of used agents is quite small. Also, no deployment of a practical application based on a prototype system is proposed. In [18], a multi-agent particle swarm optimization algorithm hybridized with a bee decision-making process is proposed to solve the power dispatch problem. The advantage of this study is that the proposed algorithm works well on different objective functions and unconstrained optimization problems. However, the convergence time increases considerably if the number of agents increases.

All the above-mentioned studies use a single hybridization scheme which is based on the simple idea of modeling the particle as an agent. Environmental modeling, particle capacity and interaction mechanisms between agents are often unspecified. In our approach, we aim to propose a more sophisticated hybridization scheme of PSO and MAS using a new concept of accent.

2.3. Other methodologies resolving the 3D deployment and their drawbacks

In this section, we detail the methods (other than optimization) used in solving the problem of 3D deployment and their complexities, advantages and disadvantages that make the optimization more appropriate to solve this problem.

2.3.1. VFA (Virtual Force algorithms)

When resolving the deployment problem, the deterministic VFA method represents the nodes as points which exert forces of repulsion and attraction between them. Nodes can move to reposition themselves thanks to the produced acceleration of these forces. As an example, [19] proposes a virtual forces approach based on a potential field deployment algorithm (PFDA) to resolve the node deployment problem. In addition, to solve the deployment in mobile sensor networks, [20] used a Delaunay triangulation relation based on the contiguity between nodes to introduce a frictional force into the Van Der Waals force equation. The most recent algorithm studying this problem is the 3D-DVFA [21]. By analyzing the latter algorithm, we highlight the major drawbacks of the VFA approach in resolving the 3D deployment

- This algorithm is based on simple (basic) repulsion-attraction movements to balance the distribution of nodes in the RoI. It tries to spread the nodes over the space without warranty of optimality or pre-optimality. It also gives a single final solution without iterating to improve the obtained solution.

- This algorithm loses its effectiveness if the nodes are heterogeneous and have different ranges of detection and transmission. These ranges are considered as deterministic (binary and spherical thus generating a sphere of detection and a sphere of transmission) which does not reflect the reality.

- This algorithm can give good results if we consider only the coverage as an objective to optimize and the connectivity as a secondary constraint to satisfy (which is equivalent to an optimization problem with one or two objectives). If the aim is to simultaneously optimize the coverage, the routing, the energy consumption and the connectivity for example, the 3D-DVFA will have difficulties in considering the mentioned objectives and its related constraints.

2.3.2. DDA and BDA algorithms

Authors in [19] propose two approximative approaches. The first one proposes a differentiated deployment algorithm (DDA) that uses 3D image modeling where the image is reconstructed using polygonal elements. DDA assumes that the mesh nodes represent the positions of the sensors and that each arc between two nodes represents the Euclidean distance between them. However,

DDA do not ensure the considered objective full network connectivity). The second approach (BDA) uses a Bernoulli random distribution that adds or removes a node in the RoI. Due to the random nature of the BDA, some cells may remain dissatisfied due to low probabilities of detection of generated events. Besides, only two objectives are considered (number of sensors and quality of monitoring).

2.3.3. Cluster-based deployment algorithms

[22] propose a deployment approach based on clustering to ensure a maximum coverage for underwater acoustic WSN. The nodes with the highest identifiers become leaders of clusters and gather the nodes that are initially randomly deployed at the bottom of the ocean. Then, these leaders identify the possible overlaps of coverage between the nodes of each cluster.

2.3.4. Geometric patterns

The geometric properties of WSNs allow using the 3D geometric structures to solve complex problems such as topology control. Recently, different based-on geometric structures algorithms are proposed for controlling the network topology.

2.3.4.1. Voronoi partition. The Voronoi Diagram subdivides the plan (the RoI in the case of 3D deployment) into a discrete set of points (sites) in such a way that each face corresponds to a region where a site is located (a node in the case of deployment of WSN). Voronoi partition is used to solve different distributed applications in WSN and robotics, such as the coverage. The Voronoi diagram of n points in the plane is calculated in a polynomial time $O(n \log n)$. Despite this reduced complexity for the case plan, the 3D case of the Voronoi partition of a region is a difficult problem due to the distributed organization of the network and the limited detection capabilities of the sensors. Moreover, in 3D deployment, the calculation of the Voronoi cell of each node, the selection of its neighbors using Euclidean distances and Cartesian positions of the nodes generate additional expensive calculations.

2.3.4.2. 3D cell partitioning. Partitioning cells is another scheme for placing nodes in WSNs. This scheme is generally controlled by the ratio between the detection radius and the communication one. The aim is to divide the 3D RoI into a set of 3D cells with a minimal number of nodes.

* Volumetric quotient-based scheme

According to [7], the most important spatial cell partitioning schemes are: cube, rhombic dodecahedron, hexagonal prism, truncated octahedron. [23] proves the efficiency of the truncated octahedron (TO) model compared to other models. However, TO requires a larger number of nodes than other models in the case of k -coverage. In this case, a compromise is made between the total required number of nodes and the radius of detection of each node.

* Grid-based deployment scheme

It is a deployment scheme in which a coordinate system is used in a 3D grid where the volume of detection is divided into a set of cubes. The size of each of these cubes is equal to the edge length of the cubic unit where the sensor space is partitioned. In this deployment model, the node fixes its depth in the grid by communicating with other nodes and by specifying the coordination distance in the neighboring cubes [7].

* Spherical overlap scheme

Spherical Cap-based Coverage This scheme assumes that the 3D field being monitored is sufficiently covered as much as the spherical detection ranges of all sensors are sufficiently covered. The problem becomes then a problem of guaranteeing that each sphere is sufficiently covered.

Reuleaux Tetrahedron-based Coverage: The spherical tetrahedron (or Reuleaux tetrahedron) is the result of the intersection of four spheres having centered vertices radii r on a regular tetrahedron having a side length r .

2.3.4.3. Continuum percolation. The theory of percolation of the continuum assumes that the nodes are scattered according to a Poisson density λ and that there is a critical density of nodes (λ_c) for which we can obtain a detectability of network near from 1. The probability of detection of an event is equal to PP_r (PP_r is a homogeneous Poisson Process in h -dimensional Euclidean Space R_h). [7] solve the problem of critical density for the percolation in connectivity and coverage in networks.

Despite nearly optimal solutions (in terms of number of nodes) generated by the geometrical deployment schemes, these latter require a relatively higher coverage compared to the communication radii of the nodes. This makes these geometrical partitioning methods ineffective for many scenarios. All shown methods suffer from the following drawbacks:

- A unique solution is provided at the end of the execution of the algorithm (and not a Pareto Front (PF) containing a set of solutions as in EMO).

- The difficulty of incorporating the constraints and the objectives and the non consideration of the many-objective case. In general, objectives are restricted to only one, or a main objective and another secondary one (for example, ensuring the coverage considering the connectivity).

- The non consideration of the user's preferences.

In this respect, an experimental study comparing all these methods with optimization evolutionary algorithms is a very interesting contribution to prove, by results, the efficiency of the latter method in resolving the deployment issues.

3. Model and mathematical analysis

In this section, we propose an analytical model for the deployment problem. The specificities of this analytical model with respect to other models can be summarized as follows: – Most deployment modelings deal with the 2D case. Little models are dedicated to the 3D case like ours.

- Unlike other formulations, the proposed modeling addresses a problematic of redeployment in a many-objective case taking into account simultaneously five antagonistic objectives: most modelings deal with a deployment problem with a main objective (such as coverage) and another (or two others) secondary objective(s) in relation with the first objective (such as connectivity or localization).

- The proposed modeling is based on objectives relying on real constraints and assumptions, likewise for architecture (based on mobile, nomads and fixed nodes) and for the type of nodes (which are used in our personal testbed). This complementarity between theoretical modeling and experimental constraints is not considered in most mathematical representations of the deployment and redeployment problems.

3.1. The network architecture

During our experimentations and simulations, the following types of node are considered:

- **Fixed nodes:** composed of the set of fixed nodes initially installed.

- **Nomad nodes:** added to improve the 3D deployment scheme. Their locations are determined by the proposed optimization algorithms. Their number depends on the simulation or experimental scenario and the size of the studied problem.

- **Mobile nodes (targets):** composed of a set of persons or mobile robots to be controlled. These targets are equipped with a radio transmitter/receiver. In the simulations, a single mobile target is used to trigger the process by sending the first message.

Table 1
Modeling parameters.

Parameter	Description
A_1	Set of possible positions where connected objects/nodes may be deployed
A_2	Set of types (fixed, nomad, mobile) of the connected objects
A_3	Set of mobile targets to be monitored
A_4	Set of connected objects having positions in A_1 and types in A_2
Tcd	Minimum number of needed nodes to cover the RoI
Cv_p^i	1 if a connected object i cover the position p , 0 otherwise
$RSig_{tgp}^i$	1 if a target positioned at $t_g \in A_3$ sends a 'receivable' signal to a connected object that is positioned at $p \in A_1$, 0 otherwise
LQI	Average-communication-quality
Av^q	Received-RSSI at a distance d from-the-sending-node $i \in A_4$
$RcRs_i^d$	1 if the object at a position $p \in A_1$ transmits a signal, with a sufficient transmission power, to another node at a site $p' \in A_1$
$TrsSg^{pp'}$	1 if the node at $p \in A_1$ receives a signal, with a sufficient transmission power, from-another-node positioned at $p' \in A_1$
$RcvSg^{pp'}$	Emitted RSSI (Transmitted signal) of the sending node $i \in A_4$
$TsRs_i^{trsm}$	Threshold for the receiver power: minimum required power of the signal (RSSI) transmitted-by-a-node-having-a-type $t_n \in A_2$ to-detect-it
$ThPw_{t_n}^{rcv}$	Lifetime of the network ($Nlf > 0$), Nlf_i is the lifetime of a node i ($Nlf_i > 0, \forall i \in A_4$). Nlf^{max} is an upper bound of Nlf and $Nlf = 1/Nlf_i$
Nlf	1 if a fixed object having a type $t_n \in A_2$ is set at $p \in A_1$, 0 otherwise
$FxN_{t_n}^p$	1 if a nomad object having a type $t_n \in A_2$ is set at a site $p \in A_1$, 0 otherwise
$NdN_{t_n}^p$	1 if a mobile node having a type $t_n \in A_2$ is set at a site $p \in A_1$, 0 otherwise
$MbN_{t_n}^p$	Upper bound of the possible number of deployed nodes in the network
N_{max}	

3.2. System modeling and notation

The 3D redeployment problem, noted **PB**, is to disseminate the connected objects (sensor nodes or things) in the three-dimensional RoI, noted **RoI** = $\{(x, y, z), 0 \leq x \leq L, 0 \leq y \leq W, 0 \leq z \leq H\}$ where L is the length of RoI, W is its width and H is its height. In the proposed architecture, a connected object may be a **fixed** one already deployed with a known position, a **nomad** one to add, (its position is determined by the optimization algorithms), or a **mobile** one attached to a target to monitor. The aim is to find the best 3D coordinates of the nomad connected objects while satisfying a set of objectives such as a maximum connectivity, network utilization and coverage guaranteeing the quality of links and the localization. **PB** is represented using the Eqs. (2), (6), (8), (11) and (13) subject to the constraints (3), (4), (5), (7), (9), (12) and (14). The used parameters are listed in Table 1.

The problem **PB** is based on the assumptions detailed in Assumptions 1–3 and the objectives in Problem 1.

Assumption 1. The number of potential locations of the connected objects is equal to the number of the used monitoring positions.

Assumption 2. The number of monitoring positions is an upper bound of the number of connected objects.

Assumption 3. Each target should be monitored by at least one connected object.

Problem 1. The many-objective fitness function is: **maximize** \vec{Y} , knowing that

$$\vec{Y} = (f1, f2, f3, f4, f5) \quad (1)$$

f1: Rate of connectivity: To consider the network as “connected”, each connected object must be able to communicate with other

ones. Thus, at least one path should be ensured from each object to another one. The probability of connectivity depends on different factors such as the number of connected objects, the received signal strength and the transmission range.

$$f1 = \text{Maximize } RcRs_i^d \quad (2)$$

$$\text{Subject to } RcRs_i^d \leq TrsSg^{pp'} * RcvSg^{pp'} * \sigma * ds^{-\lambda} * TsRs_i^{trsm} \quad (3)$$

where ds is the distance from the sending object and λ is the path loss exponent. The constraint (3) implies that $RcRs_i^d$, the received RSSI at a distance d from the sending node $i \in A_4$, is less than the emitted RSSI (the transmitted signal) of the sending node i ($TsRs_i^{trsm}$, if i receive a signal: $RcvSg^{pp'} = 1$) multiplied by the distance from the sending object (if this object transmits a signal: $TrsSg^{pp'} = 1$).

$$ds = ds_c \Leftrightarrow RcRs_i^d = ThPw_{t_n}^{rcv} \quad (4)$$

The constraint (4) implies that ds_c (representing the transmission range) is as follows: $RcRs_i^d(ds = ds_c) = ThPw_{t_n}^{rcv}$. Thus, the object sending data can send it only if the received signal is greater or equal to $ThPw_{t_n}^{rcv}$.

$$\sum_{p \in A_1} RSig_{tgp} \leq \sum_{t_n \in A_2} FxN_{t_n}^p + \sum_{t_n \in A_2} NdN_{t_n}^p \quad (5)$$

The constraint (5) delimits the number of connected objects that are able to detect a target. It is bounded by the number of the deployed objects.

f2: Utilization of the network: In general, aiming at increasing the network lifetime, many objects are installed near the base station(s), which increase costs and lead to a poor utilization of the network resources. Thus, it is recommended to maximize the lifetime and deploy a reasonable number of objects simultaneously. Increasing the lifetime can be achieved by maximizing the lifetimes of all deployed nodes and having maximum values of the variables $FxN_{t_n}^p$, $NdN_{t_n}^p$ and $MbN_{t_n}^p$. On the other hand, the sum of all lifetimes of all nodes should be less than the maximum estimated global lifetime: $\frac{\sum(FxN_{t_n}^p + NdN_{t_n}^p + MbN_{t_n}^p)}{Nlf} \leq \frac{1}{Nlf^{max}}$ which implies $\overline{Nlf} \cdot \sum(FxN_{t_n}^p + NdN_{t_n}^p + MbN_{t_n}^p) \leq \overline{Nlf}^{max}$. Thus:

$$f2 = \text{Maximize } (-\overline{Nlf} \cdot \sum(FxN_{t_n}^p + NdN_{t_n}^p + MbN_{t_n}^p)), \quad \forall p \in A_1, t_n \in A_2 \quad (6)$$

$$\text{Subject to } \overline{Nlf} \cdot \sum(FxN_{t_n}^p + NdN_{t_n}^p + MbN_{t_n}^p) \leq \overline{Nlf}^{max} \quad (7)$$

knowing that $\overline{Nlf}^{max} = 1/Nlf^{max}$

f3: Degree of-coverage: he FER metric is used to evaluate the quality of the links, which gives an idea about the degree of coverage. Indeed, less the FER is, better the coverage is. Hence, to achieve a full coverage, at least **Tcd** connected objects must be deployed to control each position p in the 3D RoI knowing that $\sum_{p \in A_1} Cv_p^i \geq Tcd, \forall i \in A_4$, which implies that $\sum_{p \in A_1} Cv_p^i - Tcd \geq 0, \forall i \in A_4$ is to be maximized for all nodes to guarantee a better localization. Thus:

$$f3 = \text{Maximize } \sum_{t_g \in A_3} (\sum_{p \in A_1} Cv_p^i - Tcd), \quad \forall i \in A_4 \quad (8)$$

$$\text{Subject to } \sum_{p \in A_1} Cv_p^i \geq Tcd, \quad \forall i \in A_4 \quad (9)$$

f4: Quality of links: The *LQI* (link quality indication) is a measure used to assess the strength of the received data packets. In general,

routes with highest overall LQI are more appropriate to deliver data to their destinations. The LQI may be estimated based on the following assumption: $LQI = \tau \times (\delta - \vartheta)$; RC is the raw LQI value calculated based on the last byte of the message. ϑ and τ are two experimental coefficients calculated according to the Packet Error Rate measures-as-a-function-of δ .

$$f4 = \text{Maximize } LQI \quad (10)$$

$$\text{Subject to } Av^{cq} = \sum_{i=1}^{N_{\max}} \sum_{j=1}^{N_{\max}} |RCRs_i^d| / \sum_{i=1}^{N_{\max}} |N_i| \quad (11)$$

The constraint (11) identifies the communication quality on the network. Based on the routing protocol, (11) proposes to compute the average communication quality (Av^{cq}) deduced by calculating the mean path-loss values of the node's direct communications. N_i is a connected object having an identifier i , $N_i \in A_4$.

f5: Rate of localization: The RSSI is used as a metric to assess the localization since the model of localization relies on a hybrid protocol (a range-free technique (3DDV-Hop) corrected by the RSSI). Thus, better the RSSI value is, better the localization is. Hence, to guarantee a better localization, at least Tcd connected objects must be deployed to control each position p in the 3D RoI knowing that $\sum_{p \in A_1} RSig_{t_g p} \geq Tcd, \forall t_g \in A_3$, which implies that $\sum_{p \in A_1} RSig_{t_g p} - Tcd \geq 0, \forall t_g \in A_3$ is to be maximized for all nodes to guarantee a better localization. Thus:

$$f5 = \text{Maximize } \sum_{t_g \in A_3} \left(\sum_{p \in A_1} RSig_{t_g p} - Tcd \right) \quad (12)$$

$$\text{Subject to } \sum_{p \in A_1} RSig_{t_g p} \geq Tcd \forall t_g \in A_3 \quad (13)$$

3.3. Relationship between the modeling and the experimentations

The relationship between theory and practice in our study is twofold and relies on the feedback between these two components: the first part concerns the use of practical findings in theory. This first part is evident in our paper, for example in the use of hypotheses and real constraints (in the mathematical modeling) derived from the testbed and the observations of the node behavior in practice. The second part concerns the use of theoretical findings in practice: indeed, theoretical results contribute in obtaining a non-basic complete testbed that uses the justified theoretical models. For example, the localization (f5 in modeling) is based on the RSSI because the localization protocol used in practice relies on DV-HOP corrected by the RSSI data. So, the higher the RSSI rate is, the better the localization is. Moreover, the coverage (f3 in modeling) is based on FER because this latter metric is used in our study to evaluate the quality of links. So, lower the rate of FER is, better the coverage is.

Besides, the used algorithmic approach is based on the concept of accent and hybridization with MAS and the modeling is used to satisfy and optimize a set of conflicting objectives such as minimizing the number of nodes and maximizing the coverage. These objectives, in addition of being antagonistic, make the deployment problem more complex and more difficult to solve by the most of methods (Virtual Forces, Voronoi Partition and MaOA (Many-objective Optimization Algorithms)). This gives the motivation to introduce modifications into MaOA and justify the theoretical hybridizations proposed to solve and reduce the complexity of such real-world complex problems.

Another motivation of the use of theoretical results in practice is that we prove in this paper the applicability of MaOAs to a real-world problem, given that most MOAs are tested on instances of theoretical problems (DTLZ, ZDT ...).

4. Resolution approach

In this section, we justify the choice than detail the use of optimization evolutionary algorithms and MAS as an approach to solve the 3D indoor deployment problem.

Indeed, the complexity (in structure and size) of the search space of each problem decides the use of heuristic resolution methods or not. If the search space is small, a deterministic method can be used, and if it is larger and more complex, it requires an approximate search method. The 3D deployment is a complex problem, proven being NP-Hard like most MaOP (Many-objective optimization problems). Exact algorithms (such as branch and bound) provide solutions for efficient deployment only if it is a small problem where the number of nodes does not exceed three. Hence, there is no exact algorithm that solves this problem in a polynomial time when the size of this problem exceeds a certain threshold. This gives the motivation and the need to use heuristic methods such as EMO algorithms to solve it.

*** Relationship between multi-objective optimization and the deployment problem:** Scheduling or planification problems (such as the deployment of nodes) are part of combinatorial optimization problems with constraints. The majority of these problems are NP-hard. Hence, one of the solutions to solve them is to use heuristics providing, within a reasonable time, a feasible and close to optimal planning. Moreover, because of the existence of a research space with a large number of solutions that are close from each other and because of the stochastic, dynamic and distributed nature of this deployment problem, evolutionary optimization is the most suitable approach to this problem.

If we increase the number of constraints (which have different contexts: from localization to routing) and the number of objectives (which are often antagonist), even EMO algorithms encounter difficulties when resolving MaOPs (such as the ineffectiveness of the recombination operation, the exponential increase of time and space costs, or the inaccuracy of the density estimation. The solution is to propose a set of well-studied and justified modifications and hybridizations of MaOAs to minimize its complexity and execution time. In our case, we proposes to modify (by adding the concept of accent birds which improves diversity and helps to escape optima locales) and hybridize (with MAS for more interaction and better performance) the MOPSO struggling when resolving the many objective 3D deployment problem.

*** Relationship between MAS and the deployment problem:** The use of MAS is also justified by the nature of the deployment problem (a problem of a distributed nature) and the nature of the environment of this problem (complex and highly dynamic). Moreover, the choice of MAS is justified by the existence of sub-tasks (nodes, sub-regions to cover, etc.) and a high communication between entities (nodes). Despite this efficiency, the 3D deployment problem is not well investigated using meta-heuristics and EMO algorithms. Thus, another motivation of the use of EMO (and MAS), is to prove its efficiency in solving complex real world many-objective problems. In addition, unlike geometric approaches (Cell partitioning, Voronoi Partitions) solving the deployment problem, the optimization by meta-heuristics gives the user the ability to choose between several solutions and guide the search towards the set of desired solutions.

4.1. Many-objective PSO algorithm

Proposed by [24], 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

$$\vec{\mathbf{A}}_{1_i}(t) = \vec{\mathbf{A}}_{1_i}(t-1) + \vec{\mathbf{M}}\mathbf{v}_i(t) \quad (14)$$

and a speed of movement noted

$$\begin{aligned} \vec{\mathbf{M}}\mathbf{v}_i(t) = & \omega \vec{\mathbf{M}}\mathbf{v}_i(t-1) + c_1 r_1 (\vec{\mathbf{A}}_{1_pbest} - \vec{\mathbf{A}}_{1_i}(t)) \\ & + c_2 r_2 (\vec{\mathbf{A}}_{1_gbest} - \vec{\mathbf{A}}_{1_i}(t)) \end{aligned} \quad (15)$$

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 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 [25], 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 [26] proposed an algorithm that empowers the multi-objective structure of the PSO to deal with many-objective problems and suggest a R_2 indicator to guide the search. In [27], the authors proposed a based-on archiving PSO algorithm named I-MOPSO which explores specific aspects of the MOPSO to handle many-objective problems while introducing more convergence and diversity on the search.

In [28], 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.

4.2. Multi-agent systems

A MAS is 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).

4.3. A hybrid modified MaOPSO-MAS algorithm

4.3.1. The proposed acMaPSO algorithm: including the concept of bird's accent on the 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.

- The bird accent concept

Indeed, according to recent research in biology [29], 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 Fig. 1, 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 ($\mu \ll \varphi$; $\varphi = \beta \times \mu$, $\beta > 5$), the clusters (swarms) are updated to allow the search to escape from local optima.

- The accent measure

Learning accents in particles reflects the number of iterations after which the *pbest* position of a particle has not been updated. 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.

- 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

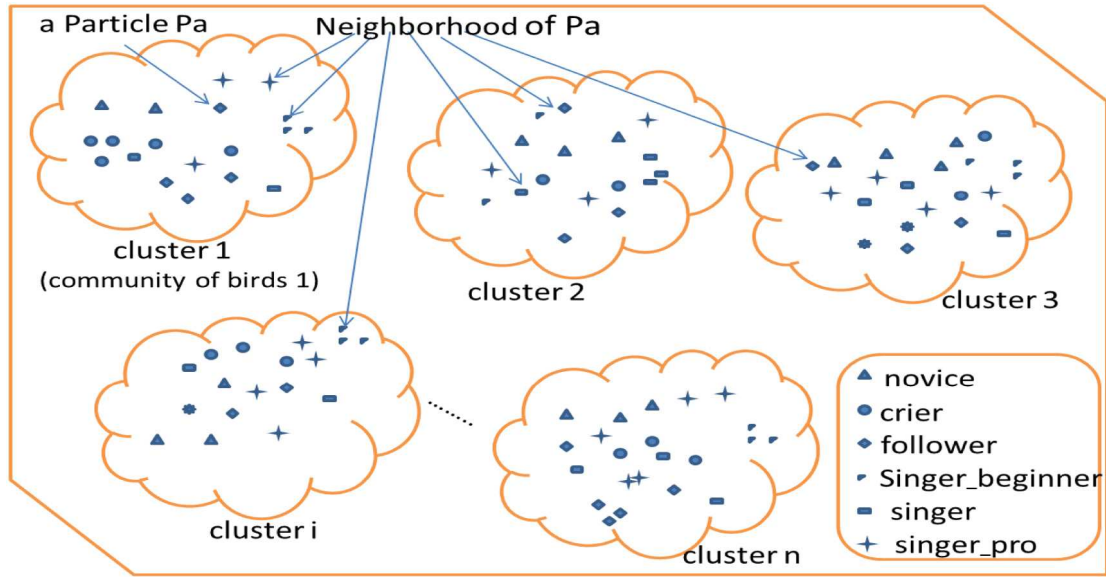


Fig. 1. The choice of the neighborhood of a particle P_a .

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 these 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 communitybest) to keep the best position of the cluster.

Algorithm 1 illustrates the acMaPSO algorithm.

4.3.2. 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 [30]. In addition, 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 (its computing is time-consuming), 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: for 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.

*Capabilities of the agents in our MAS

```

Initialize the parameters  $n$  (number of swarms),  $k$  (population size of each swarm),
 $w$  (inertia weight),  $c1$  and  $c2$  (acceleration coefficients)
Set the fitness evaluations counter  $C_f = 0$ 
Set the initial accent  $a_i$  of initial swarms to novice
Random generation of  $n$  initial swarms  $s_i \in S$  with size  $P$ , ( $i = 1, 2, \dots, P$ )
for  $i=1$  to  $n$  do
  Initialize parameters of the swarm  $i$ 
  for  $j=1$  to  $k$  do
    Initialize position and velocity
    Calculate fitness
  end for
end for
while (stop condition is not fulfilled [sufficient good fitness | maximum number of iterations]) do
  if highest_experience_category is reached then
    Delete particles having reached the highest experience category and insert new random particles (accent operator)
    Construct accent-group topology structure (S,a)
  end if
  if (fitness if not evolved in recent  $\mu$  iterations) then (move on to the next category)
  end if
  for  $i=1$  to  $n$  do
    for  $i = 1 : P$  do
      Adjust the inertial weight (S; w; a)
      Adjust the acceleration constant (S; c1; c2; a)
      Update position and velocity using equations (14) and (15)
      Evaluate the fitness (S,  $C_f$ )
      if ( $f_p$  is better than  $f(pBest)$ ) then ( $pBest = p$ )
      end if
    end for
     $cBest_i = \text{best } pBest_i$  in  $P$  //updating  $cBest_i$  (the  $cBest$  of the swarm  $i$ )
  end for
  Update accent of particles
   $gBest = \text{best } cBest_i$  in  $P$  //updating  $gBest$  (the  $gBest$  of all the swarms)
end while

```

Algorithm 1. The acMaPSO algorithm

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 [15], 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. The main features and capabilities of the agents in our MAS are as follows

– **Search local ability:** concerns the analysis, when flying, of the local distribution of the search space. This concerns, for example, identifying and memorizing the most promising flight directions.

– **Self-locating:** each agent can locate itself in the search space (determine if its solution for the deployment is feasible or not for example).

– **Life-cycle of an agent:** The closer the agent is from the flat area of the search space, the more likely it dies. On the other hand, the closer the agent is from the overall optimal agent, the more it survives and the more it attracts others agents.

– **Learning:** An agent can learn from its environment and from other agents. This process is manifested in the analysis of the relationship between speed, position and migration gain (change of community). This learning ability may concerns also the energy cost of moving.

– **Communication:** when flying, the agent-particle communicates with other agents, retrieves information concerning the velocities and positions of other agents, and whether their positions are better than its position.

* **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 updates its position and velocity. The overall procedure is illustrated in Fig. 2.

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 Eq. (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. 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:

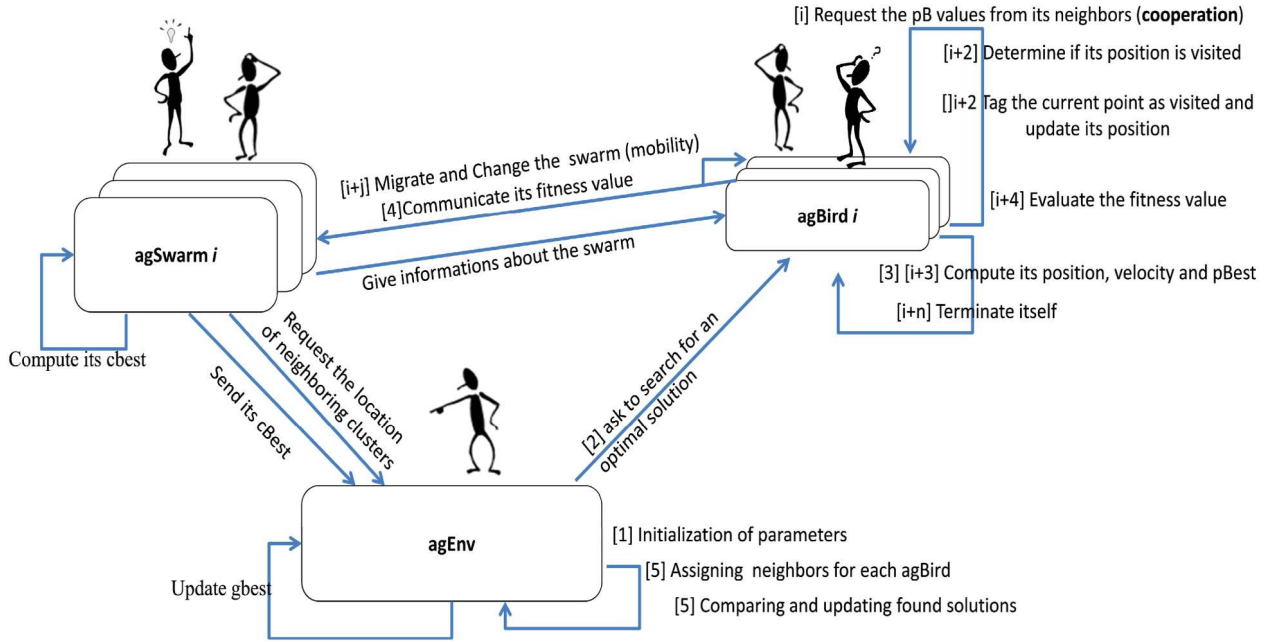


Fig. 2. The proposed MAS architecture.

- n which is the dimension of the problem space.
- P which is a set of points in the problem space knowing that the position $p \in P$ and $\mathbf{p} = \{d_1 m \times d_2 \times \dots \times d_n, visited\}$ 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.

- $\mathbf{C} = \{(p_1, p_2, \dots, p_{center}, p_m), d \mid d \geq \epsilon_d\}$ which represents the set of clusters. p_{center} is the center point of the cluster. d is the current density of the cluster. ϵ_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 Ag be the agent bird (or the agent particle) located in the position p , 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

Indeed, to create more dynamism and to allow the migration of agents (to move between communities of birds and learning new accents), a mechanism of negotiation between $agSwarm$ and $agBird$ agents is proposed. $agSwarm$ can trigger this negotiation by proposing to the $agBird$ agent having the best objective function value to migrate. The latter can accept or refuse according to a random function that is based on the result of the following question that asks to its neighbors *do you see that it is useful for me to leave my current community?* This $agBird$ follows the dominant decision of its neighbors. Nevertheless, a set of neighboring $agBird$ agents can trigger a communication between themselves to move the smallest fitness agent. The process of negotiation and communication between agents is as follows: Each agent is in conflict or in cooperation with its neighbors depending on its local environment and intentions. Let $\mathbf{Loc} = (l_1, l_2, \dots, l_n)$ be the set of possible locations in the range of the optimization solution (\mathbf{Loc} is different from A_1) and $\mathbf{Ag} = (Ag_1, Ag_2, \dots, Ag_n)$ the agent having the smallest fitness between four neighboring agents. \mathbf{Loc} is considered a winner only if its fitness function is less than the fitness function of \mathbf{Ag} . In this case, its position in the solution space remains the same. Otherwise, its position will be modified.

5. Results

5.1. Numerical results

In this section, the algorithms parameters and the obtained numerical results are presented. Different performance metrics can be used to compare the approximations sets produced by running a MaOA using numerical values:

Indeed, the hypervolume (HV) [31] is a unary metric, also called the *hyper-zone S* metric or the *measurement of Lebesgue*. It represents the volume of the objective space dominated by a set of solutions. The HV can evaluate different aspects such as precision, diversity and cardinality. In the case of maximization of objectives

Table 2
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 3		
Maximum number of generations	Variable, see Table 3		
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 Eq. (16)	
	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

(such as ours), higher the HV value is, better the quality of the solution is.

The inverted generating distance (IGD) is another metric which represents the inverted variation of the GD metric. The difference is that the IGD calculates the minimal Euclidean distance between a set of approximations and the PF, while the GD calculates the average distance between them. Moreover, to calculate the distance between two sets, IGD uses the PF solutions as a reference instead of solutions in the approximation set. According to [32], if we know a sufficient number of PF members, IGD can simultaneously measure the diversity and the convergence of a set of approximations.

Hence, what measures must be chosen to evaluate the performance of MaOPs?: Indeed, for large dimension PF for example, calculating the HV is more expensive than calculating the IGD. However, the *reference set* in HV is easier to build than the *comparison set* in IGD. If we know the true PF of the problem, we can measure the proximity with this the real PF. Otherwise, the space metric or the HV can be used. The latter, despite its cost, it evaluates the propagation and convergence of solutions at the same time, and it is ideal for real world with unknown PF. In our case, we use HV to evaluate the behavior of the algorithms on the real problem of deployment and the IGD to evaluate them on DTLZ1-4 test problems.

5.1.1. PSO and NSGA-III parameters

The used setting parameters are detailed in Table 2.

The sum of cognitive and personal components is generally set to four as recommended in [33]. In [34], 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 [33], 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

$$5 - 4 * (it/tIT) \quad (16)$$

where it is the current iteration and tIT is the maximum number of iterations.

5.1.2. 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 3 demonstrates the obtained HV values for different numbers of objectives and generations. Best performances (having higher HV) are shown with shaded backgrounds.

Obtained results show that for different numbers of objectives, acMaPSO and acMaMaPSO are often more efficient than other algorithms. Moreover, although MaOPSO is more efficient than NSGA-III, it has higher relative degradation when the number of objectives increases.

5.1.3. Results of DTLZ test problems

To prove its scalability, we test the performance of our approach on the theoretical test suite (DTLZ [35]) 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 [36] metric and 15 independent runs in each test. IGD allows measuring both the diversity and the convergence of the non-dominated obtained solutions. Table 4 shows best, average and worst IGD values of the DTLZ1-4 test suite. Best performances (having lower IGD values) are shown with shaded backgrounds.

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.

5.2. 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. In the following, we cite the advantages of using a personal testbed like our testbed (we call it Ophelia):

- **Realism:** Ophelia is based on real testbeds that reflect a real use context, contrary to the FIT/IoT-LAB testbed approach for example, which implements a very large number of nodes but are uniform and aligned with a grid.

- **Reproducibility:** Ophelia is a low-cost, open source-based testbeds based on free tools (OpenWiNo and Arduino). This allows other research teams to handle and reproduce it easily. Indeed, compared to other platforms, Ophelia supports several physical layers and integrates easily different sensors, which allows research teams to quickly and easily deploy their testbed into real environments.

- **Heterogeneity of nodes:** three types of nodes (TeensyWiNo, DecaWino and WiNoLoRa) are supported by Ophelia. Since this latter is compliant with the open hardware and software world, the WiNo architecture allows adding foreign libraries in relation to the deployed nodes. This allows adding a large variety of nodes.

- **A distributed deployment site:** Ophelia is based on nodes that are deployed on a 200 * 200 m² campus with a distribution of the nodes on several buildings of the site (departments, offices).

- **Ease of use and deployment:** Nodes are easy to deploy and manipulate (erasing data, carrying-out), which allows customizing them to design and implement a realistic and simple network. Indeed, using OpenWiNo, the deployed nodes execute the protocol stack via their wireless interface. Then, they are managed via their USB interface using the kernel console. The physical layer is simple

Table 3
Hypervolume values (Best, average and worst) on the deployment problem.

ObjNbr	Max nbr of generations	NSGA-III	MaOPSO	acMaPSO	acMaMaPSO	MOEA/DD	Two_Arch2	R ₂ MPSO
3	1300	0.902231	0.903458	0.903631	0.903774	0.902892	0.901832	0.901432
		0.901658	0.902896	0.903036	0.903039	0.902841	0.901827	0.901231
		0.898235	0.898023	0.902563	0.901088	0.902836	0.901311	0.901034
4	1800	0.974892	0.976985	0.977331	0.977329	0.974694	0.976493	0.973489
		0.974743	0.976833	0.977098	0.977243	0.974527	0.974512	0.972632
		0.973897	0.975612	0.976892	0.976838	0.974512	0.973629	0.972296
5	2600	0.972983	0.972892	0.972985	0.973324	0.972762	0.973334	0.972929
		0.972563	0.972116	0.972728	0.972984	0.972689	0.972663	0.972641
		0.972126	0.971084	0.972636	0.972356	0.972537	0.972607	0.972173

Table 4
Best, average and worst IGD values on DTLZ1-4 test suite.

Test Pb	ObjNbr	Number of generations	NSGA-III	MaOPSO	acMaPSO	acMaMaPSO	MOEA/DD	Two_Arch2	R ₂ MPSO
DTLZ1	3	400	4.880 E-4	3.458 E-4	3.566 E-4	3.238 E-4	3.191 E-4	1.089 E-3	4.123 E-4
			1.308 E-3	1.231 E-3	1.209 E-3	1.298 E-3	5.848 E-4	1.228 E-3	1.514 E-3
			4.880 E-3	4.783 E-3	3.844 E-3	4.257 E-3	6.573 E-4	1.523 E-3	4.629 E-3
	5	600	5.116 E-4	0.635 E-4	0.642 E-4	0.637 E-4	2.635 E-4	6.034 E-4	3.264 E-4
			0.980 E-4	0.684 E-4	0.649 E-4	0.648 E-4	2.916 E-4	6.372 E-4	6.487 E-4
			1.979 E-3	1.323 E-3	1.589 E-3	1.635 E-3	3.109 E-4	6.967 E-4	1.798 E-3
DTLZ2	3	250	1.262 E-3	5.663 E-4	3.856 E-4	5.734 E-4	6.684 E-4	6.884 E-4	5.237 E-4
			1.357 E-3	6.102 E-4	5.897 E-4	5.023 E-4	8.073 E-4	7.122 E-4	6.529 E-4
			2.114 E-3	7.029 E-4	8.014 E-4	6.298 E-4	1.243 E-3	7.697 E-4	8.136 E-4
	5	350	4.254 E-3	1.302 E-3	1.279 E-3	1.243 E-3	1.428 E-3	1.203 E-3	1.428 E-3
			4.982 E-3	1.534 E-3	1.432 E-3	1.412 E-3	1.491 E-3	4.760 E-3	1.443 E-3
			5.862 E-3	1.896 E-3	1.982 E-3	1.786 E-3	1.494 E-3	4.992 E-3	1.644 E-3
DTLZ3	3	1000	9.751 E-4	7.235 E-4	6.088 E-4	6.320 E-4	6.690 E-4	3.237 E-3	9.631 E-4
			4.007 E-3	3.028 E-3	3.011 E-3	2.875 E-3	2.892 E-3	3.742 E-3	3.567 E-3
			6.665 E-3	6.657 E-3	6.234 E-3	4.657 E-3	6.231 E-3	4.723 E-3	9.094 E-3
	5	1000	3.086 E-3	1.432 E-3	1.533 E-3	1.512 E-3	2.181 E-3	4.364 E-3	1.998 E-3
			5.960 E-3	2.569 E-3	2.167 E-3	2.247 E-3	3.365 E-3	4.834 E-3	2.337 E-3
			1.196 E-2	6.233 E-2	5.798 E-3	6.064 E-3	6.736 E-3	6.230 E-3	6.089 E-3
DTLZ4	3	600	2.915 E-4	2.846 E-1	4.321 E-1	3.654 E-1	3.025 E-4	5.236 E-4	2.976 E-1
			5.970 E-4	6.280 E-1	7.194 E-1	6.025 E-1	6.429 E-4	5.984 E-4	4.235 E-1
			4.286 E-1	6.685 E-1	8.632 E-1	6.309 E-1	6.881 E-4	6.382 E-4	5.249 E-1
	5	1000	9.849 E-4	3.093 E-1	6.058 E-2	8.052 E-2	9.997 E-4	9.932 E-4	1.096 E-1
			1.255 E-3	4.834 E-1	0.381 E-1	0.352 E-1	1.296 E-4	1.225 E-4	5.562 E-1
			1.721 E-3	4.992 E-1	2.984 E-1	2.554 E-1	1.532 E-4	1.687 E-4	7.443 E-1

to replace because the WiNo nodes are compatible with both standardized physical layers and groundbreaking transmission modes such as UWB and LoRa.

– **Real-life usage:** WiNos nodes are ideal for prototyping communicating objects due to their small size and low energy consumption. They are also easy to attach to a person or a mobile system. This makes them an ideal component of the IoT.

5.2.1. Description of the testbed and the used tools

The deployed TeensyWiNo nodes. In this study, we used TeensyWiNo nodes which are based on the WiNoRF22 nodes equipped with temperature and brightness sensors, to which sensors of pressure, acceleration, compass and gyrometer are added. Indeed, TeensyWiNo nodes are designed to provide access to the low layers for a developer who wants not only to have a precise control over the access time to the medium, the sleep time and the node awakening but also over the CPU time and the management of the restricted memory. Whether for the purpose of controlling drastic energy-saving policies or for respecting real-time constraints, such control of all the components of the node is necessary. TeensyWiNo nodes are a hardware platform which is a candidate for hosting protocols with high temporal constraints (several months of operation using two AAA batteries). Table 5 summarizes the technical characteristics of the installed TeensyWiNo nodes.

The components and an example of the deployed TeensyWiNo nodes are shown in Fig. 3.

Table 5
TeensyWiNo technical characteristics.

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

The TeensyWiNos are integrated into the Arduino ecosystem, enabling researchers to easily integrate hardware and software bricks (sensors, actuators, processing algorithms, interaction devices, etc.) and prototyping solutions that allow the feedback of the user experience.

Arduino [37] is an open software/hardware platform used to prototype modules. WiNo nodes use it to transfer the programs using serial links. Teensyduino is an Arduino add-on used to run sketches.

OpenWino (Open Wireless Node) [38] is a free development environment for protocol engineering in WSN and DL-IoT. It allows the rapid prototyping of protocols (MAC, NWK and other layers) and the pragmatic evaluation of their performances, in C language. It also allows them to be run on testbeds, via real nodes (WiNos) which rely on the Arduino environment and are developed in Open Hardware. Coupled with OpenWiNo software, WiNo nodes form a self-organized mesh network which is open protocol.

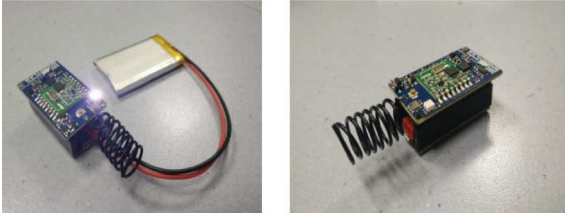


Fig. 3. The Teensy WiNo deployed nodes.

Table 6
Experimental/simulation parameters.

Number of nodes	36 (29 fixed, 6 nomads, 1 mobile)
Repartition of nodes	6 sites on 200 * 200 m ²
Transmission power	100 mW
Reception gain	50 mA
Operating temperature	25 °C
Bit rate	256 kbps
RSSI (Received Signal Strength Indicator)	100 (initially)
FER (Frame Error Rate)	0.01 (initially)
Indoor transmission range	7 m
Indoor sensing range	8 m
Antenna model	transceiver RFM22
Modulation model	125 kbit/s GFSK
Frequency	434.79 MHz
Average of runs	25 experiments
Modem configuration	12 # GFSK_Rb2Fd5
Message-wait	5
Message-number	1000
Message-length	16
Tx power	7 (the max of RFM22)
Simulation period	10800 s

This approach allows a great versatility on the hardware. Thus, OpenWiNo, associated with the WiNo hardware, is advantageous compared to other platforms and testbeds. The main advantage of OpenWiNo is its simplicity. For example, on a WiNo, to change the Physical layer, 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, among them: Classical IEEE 802.15.4 2.4 GHz DSSS (Freescale), IEEE 802.15.4-2011 UWB (DecaWave DW1000), LoRa mode 868 MHz (HopeRF RFM95) and Proprietary 433 MHz FSK/GFSK (HopeRF RFM22b).

Ophelia. Ophelia is a platform which is based mainly on Arduino, openwino and the installed Teensywino nodes, to which a web interface is added. This interface allows remote access to Ophelia testbed and the programming and execution of experimental sketches (in python language) on the nodes. In the following, we detail the execution scenario of our experiments using Ophelia.

Concerning the simulations, it is carried out using OMNeT++ which is a platform for simulating and developing network protocols. OMNeT++ IDE contains a topology description language called NED, a simulation kernel library and a graphical user interface for running the simulation. Fig. 4 demonstrates the indoor deployment scheme of the nodes in one site (of the six sites).

Fig. 5 shows an OMNeT++ simulation interface that represents the distribution of the nodes.

5.2.2. Experimental/simulation parameters

The algorithms are tested on an Intel Core i5-6600K 3.5 GHz computer. Our model implements a 433 MHz physical layer, a non-coordinated IEEE 802.15.4 CSMA/CA access method and an AODV protocol as a routing layer. The parameters in Table 6 are used in our experiments and simulations.

5.2.3. Comparing the simulations to the experimental results on Ophelia testbeds

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 its 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 another node j if the average of the RSSI emitted from i to j and the one from 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 algorithms (acMaPSO and acMaMaPSO), these latter are compared to the NSGA-III [39], MOEA/DD [40], Two_Arch2 [41], R₂MPSO [26] and MaOPSO [25] which are recent many-objective optimization algorithms. 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-node i . Once i 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.

Comparing the RSSI rates: To assess the connectivity ($f1$ in our modeling (see Section 3.2)), the quality of links ($f4$) and the localization ($f5$), the RSSI metric is used. It is a measure indicating the power (intensity) of a signal (usually a radio signal) received from an antenna. RSSI is used since localization is based on RSSI and Distance-VectorHop protocol. So, the higher the RSSI rate, the better the localization. Fig. 6 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.

Fig. 6 shows first of all a similarity in the results of the simulations and those of the experiments considering each algorithm alone. This is due to the fact that simulation and experimentation scenarios are similar, and also because the simulation uses (initially) the RSSI and FER matrices of the experiments.

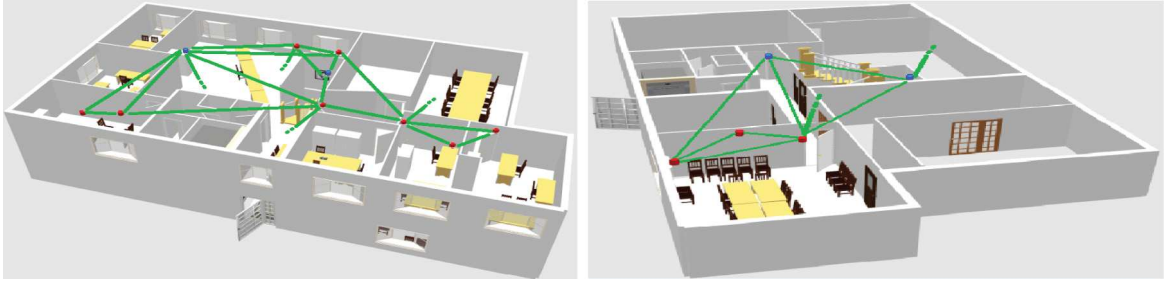


Fig. 4. 3D deployment scheme of the real experiments in one of the six sites.

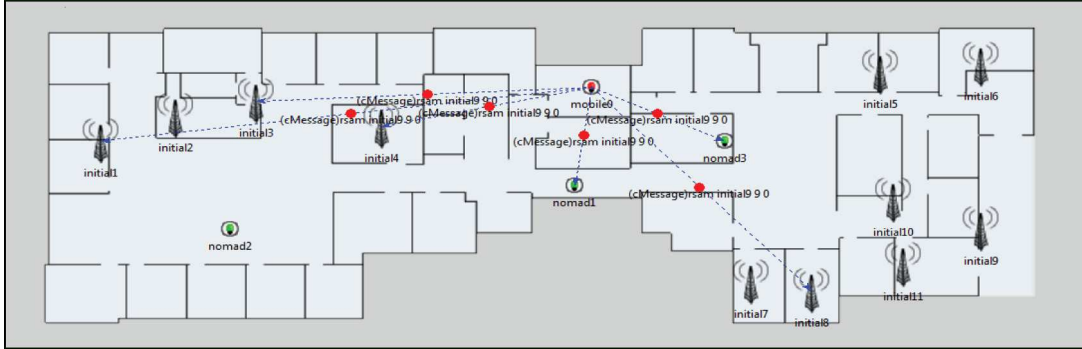


Fig. 5. The simulation scenario.

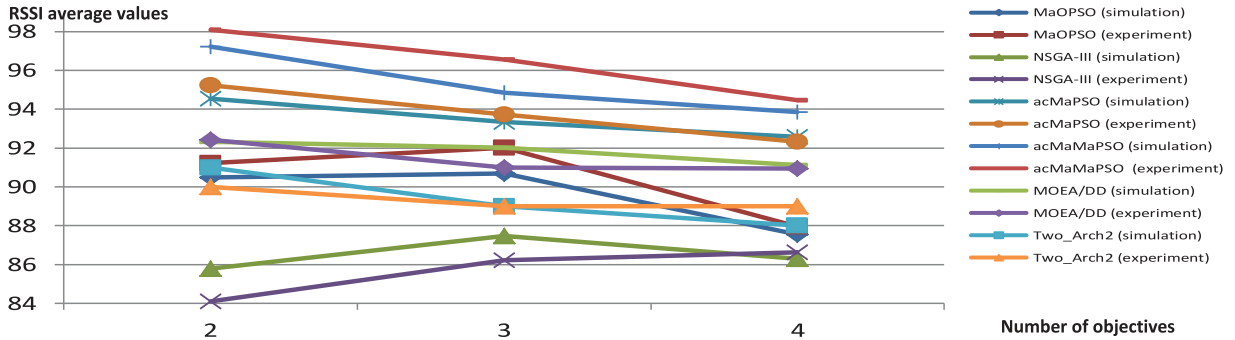


Fig. 6. RSSI average rates of nodes in connection with the mobile node.

Another finding is that the highest RSSI rates were recorded for acMaMaPSO (then, for acMaPSO) and the lower RSSI rates concern the NSGA-III in experimentation and simulation. This indicates the high performance of acMaMaPSO and acMaPSO compared with other algorithms. In addition, it is proven from Fig. 6 that the efficiency of algorithms depends on the number of objectives: for example, NSGA-III is more efficient for three or more objectives. This fact confirms, in a real world context, the assertion of the NSGA-III authors indicating that it is not dedicated for single and two-objective problems.

Comparing the FER rates: To measure the coverage (f_3), FER is used as a metric. It is a measurement used to test the performance of a link between two nodes based on the ratio of data received with errors compared to the total received data. FER is used to evaluate the quality of links between nodes. Thus, the lower the FER, the better the coverage. Fig. 7 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.

Contrary to RSSI, the FER rates are more important in experiment than in simulation. Almost the same RSSI findings are identified for the FER as regard the influence of the number of objectives on the behavior of the algorithms and the efficiency of the acMaMaPSO and acMaPSO compared to other algorithms.

Comparing the number of neighbors: To measure the network connectivity (f_1) and the utilization of the network (f_2), 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. Fig. 8 illustrates, for different numbers of objectives, the average number of neighbors measured for all nodes in connection with the mobile node.

Fig. 8 also asserts that acMaMaPSO (in simulations) (then, acMaPSO in simulations) gave the best quality of solution in terms of the average number of neighbors per node.

Comparing the energy consumption and the network lifetime Fig. 9 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.

The Fig. 9 indicates that acMaPSO (in experiments) gives the best energy consumption levels. After the acMaMaPSO, it is the

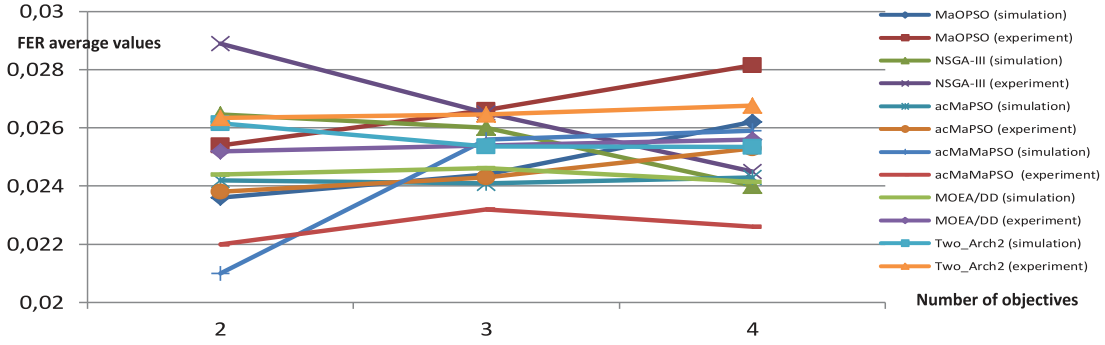


Fig. 7. FER average rates of nodes in connection with the mobile node.

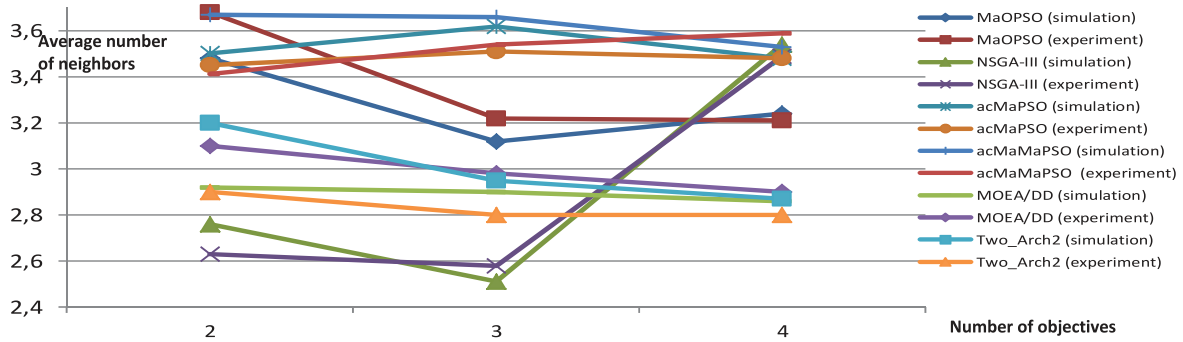


Fig. 8. Average number of neighbors of nodes in connection with the mobile node.

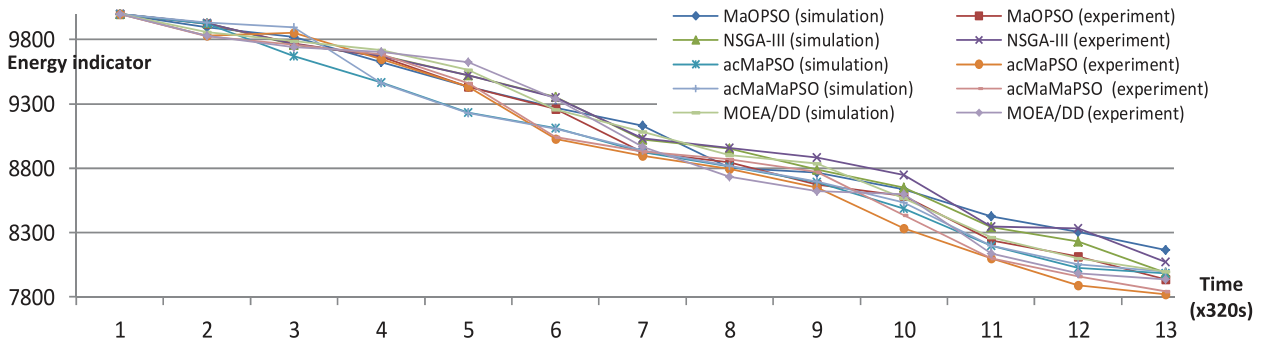


Fig. 9. Comparing the average energy consumption levels.

acMaMaPSO which is better than the rest of algorithms. The worst energy consumption levels were recorded for the NSGA-III.

Table 7 shows the lifetime of the network. It illustrates for two numbers of objectives, the time in which the first node is dead.

Comparing the execution time, according to Table 7, the best algorithm is the acMaPSO while the worst one is acMaMaPSO. This is explained by the advantage of acMaPSO compared to other algorithms (as explained before in Section 4.3) and the additional costs in time of the acMaMaPSO caused by inter-agent communications.

5.2.4. 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.

- The superior performance of the acMaPSO as regard other algorithms can be explained by the concept of autonomous self-population maintenance where a particle in this new multi-population swarm (based on accent birds) has the ability to maintain the subpopulation it lies in. For example, it can find the best sub-population distribution, the best sub-population size and different other features.

- In agreement with the findings of different other studies (such as [42]) 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.

Table 7

Comparing the average lifetime.

Nbr of objectives	MaOPSO		NSGA-III		acMaPSO		acMaMaPSO		MOEA/DD		Two_Arch2	
	s	e	s	e	S	e	s	E	s	e	s	E
2	3546	3502	3485	3487	3543	3549	4572	4583	3589	3623	3632	3669
5	3478	3469	3528	3546	3540	3553	4568	4571	3895	3934	3798	3831

s: simulations e: experiments.

Table 8

Obtained average errors values for DTLZ1-4.

	NSGA-III	MOEA/DD	Two_Arch2	R ₂ MPSO	MaOPSO	acMaPSO	acMaMaPSO
DTLZ1	2.319 10 ⁻²	1.985 10 ⁻¹	3.568	1.942 10 ⁻¹	2.973 10 ⁻²	2.035 10 ⁻²	3.564 10 ⁻²
DTLZ2	1.526	0.0000	8.197 10 ⁻⁴	3.205 10 ⁻³	6.981 10 ⁻²	3.205 10 ⁻³	0.0000
DTLZ3	4.984 10 ⁻⁴	3.984 10 ⁻⁴	2.039 10 ⁻³	6.197 10 ⁻²	7.0292	6.197 10 ⁻²	5.168 10 ⁻²
DTLZ4	3.028 10 ⁻³	5.194 10 ⁻²	0.0000	4.192	9.119 10 ⁻⁴	4.192 10 ⁻²	4.069 10 ⁻²

– 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 algorithms, the NSGA-III in simulations has higher RSSI than in experiments.

– In comparison with RSSI, the FER rates are more important in experiments than in simulations.

5.3. Statistical nonparametric tests for comparison between algorithms

In this section, we evaluate the competitiveness of acMaPSO and acMaMaPSO from a statistical point of view. Indeed, in addition to the numerical tests with the metrics of performance evaluation (HV and IGD) and the experimental metrics (RSSI, FER, Number of neighbors), performing a statistical test is very useful to prove the robustness of the obtained results regarding the behavior of the proposed algorithms. Generally, we use statistical tests to evaluate one (or more) population(s) through samples of data. We choose to perform a pairwise test (wicolxon signed ranks test) and a multiple statistical comparison test (Friedman Test).

Indeed, in statistics, such statistical tests imply the preliminary definition of two hypotheses. A first (called null hypothesis H_0) indicating that there is no difference between the compared populations, and a second (called alternative hypothesis H_1) indicating the existence of a difference between compared populations (then, between compared algorithms). The application of such a test with hypothesis requires the definition of a level of significance (α) to identify the level in which the hypothesis can be discarded.

For both tests (wicolxon and Friedman), DTLZ test problems were used with 4 objectives, 9 variables for DTLZ1 and 11 variables for DTLZ2-4. The values of the parameters of the used algorithms are the same as in simulations and experiments (Section 5.2). Table 8 illustrates the average errors values (for 30 runs).

5.3.1. Pairwise comparisons test: Wilcoxon signed ranks tests

The Wilcoxon signed ranks, considered as the non-parametric equivalent of the parametric t-test, relies on the pairwise comparison of different paired groups (algorithms in our case), to assess their differences in behavior. We use this test to decide to reject or accept the null hypothesis asserting that two different algorithms represent two distinct populations. Initially, we calculate the signs R^+ and R^- between acMaPSO and each other algorithm. Then, their associated p-values are determined, while ensuring the following property $R^+ + R^- = n(n+1)/2$ for compared each pair. Table 9 shows the computed R^+ , R^- and p-values for pairs of algorithms using the Wilcoxon Test.

The obtained results confirm that acMaPSO has an improvement over NSGA-III, with a level of significance $\alpha = 0.1$, over

Table 9Computed R^+ , R^- , and p-values for all pairs (Wilcoxon test).

	acMaPSO - NSGA-III	acMaPSO - MOEA/DD	acMaPSO - Two_Arch2	acMaPSO - MaOPSO	acMaPSO - R ₂ MPSO
R^+	282	265	273	289	236
R^-	79	54	32	82	87
p-values	0.0741	0.0063	0.0085	0.0234	0.0418

Table 10

Average ranking (Friedman test).

Test Pb/ algorithm	NSGA-III	MOEA/DD	Two_Arch2	acMaPSO	MaOPSO	R ₂ MPSO
DTLZ1	3	2.5	2	2	3	3
DTLZ2	3	2.5	3	1.5	2.5	2.5
DTLZ3	3	3	1.5	1.5	3	3
DTLZ4	4	4	4	2.5	4	3
Average	3.25	3	2.625	1.875	3.125	2.875

MaOPSO and R₂MPSO, with $\alpha = 0.05$, and over MOEA/DD and Two_Arch2 with $\alpha = 0.01$.

In what follows, we compare simultaneously several algorithms together using the Friedman Test which considers a control method for these comparisons. In order to compare the performance of the new proposed algorithm with other state-of-the-art algorithms, this new algorithm is generally taken as the control method.

5.3.2. Multiple comparisons test: the Friedman test

The Friedman test [43], considered as the nonparametric equivalent of the repeated measures ANOVA test, relies on a multiple comparisons for computing the groups differences using a control method. In Friedman's test, the null hypothesis assumes that the medians between populations are equal while the alternative hypothesis considers the opposite case. This test is used to know if at least two of the set of samples represent populations with different median values. This test relies on converting the initial results into ranks. Table 10 illustrates the average ranking of the acMaPSO algorithm (our control method) and others algorithms with $\alpha = 0.05$.

Obtained results show that, according to the ranking, NSGA-III is the worst algorithm (highest rank) while acMaPSO is the best (lowest rank). This is congruent with the results of Wilcoxon test accepting the null hypothesis and assuming that acMaPSO is better than other algorithms. This performance of acMaPSO can be explained by the enhancement of the diversification introduced by the bird's accent concept (compared to MaOPSO, R₂MPSO and other algorithms).

5.3.3. Post-hoc procedures

Despite its effectiveness, the Friedman test is unable to make appropriate comparisons between the considered algorithms. It

Table 11
Results of the Post-hoc procedures (Friedman test).

	NSGA-III	MOEA/DD	Two_Arch2	MaOPSO	R ₂ MPSO
Hochberg	0.0265	0.000502	0.000312	0.00359	0.009688
Li	0.000698	0.000132	0.000481	0.000982	0.000667
Holm	0.00951	0.00783	0.00634	0.00223	0.0037
Holland	0.000251	0.000494	0.000293	0.000378	0.000324
Unajusted	0.0102	0.00023	0.0085	0.0234	0.0418

has the disadvantage of only detecting significant differences across the entire multiple comparisons. If multiple tests aim to perform a comparison by considering a method of control and a set of algorithms, we can define a family of hypotheses that are related to the method of control. Afterwards, a post-hoc test is used to obtain a p-value that determines the degree of rejection of each hypothesis. A family of hypotheses is composed of a set of logically comparable interdependent hypotheses. If we have Q algorithms, in $1 \times Q$ comparisons, the control method is compared with the other $k-1$ study algorithms. In the $Q \times Q$ comparisons, we have $k(k-1)/2$ possible comparisons between the different algorithms. Thus, we obtain a family of $k-1$ or $k(k-1)/2$ of hypotheses, respectively, which are ordered by their p-value, in increasing order.

In our case, the p-value is set to 0.002 (calculated by the Friedman Test). Since $p\text{-value} < \alpha$, the Friedman test rejects the null hypothesis (Equality of medians). Hence, there is a difference between the algorithms. Following this rejection of the null hypothesis, we can perform a post-hoc test to control the family-wise error. We can use post hoc procedures such as Hochberg, Li or Holm. Table 11 illustrates the p-values of these post-hoc procedures (acMaPSO is the control method).

Obtained results show that acMaPSO is more competitive than other algorithms for different post-hoc procedures. The best behavior (lowest p-values) was recorded for Li and Holland tests.

Nevertheless, the Friedman test is based on a ranking system that allows only intra-set comparisons. This presents a drawback if the number of algorithms to be compared is small because the intra-set comparisons may not be significant. In this case we can use other more performing tests like Quade tests or Friedman Aligned Ranks.

5.4. Computational complexity and runtime analysis

A study of the temporal complexity is very useful in evaluating the behavior of the proposed algorithms with respect to other algorithms in terms of computing time.

* Computational complexity of algorithms

The computational complexity indicates the rate of increase in execution time proportionally to the population. Low time execution does not indicate a low calculation complexity. Table 12 illustrates the computational complexity of different algorithms. Each algorithm is tested on the DTLZ1 problem with a population of n individuals.

Table 12 shows that MaOPSO/2s-pccs has the highest computational complexity and that acMaPSO and MOPSO have the lowest complexity. The complexity of acMaMaPSO is higher than that of acMaPSO because of the additional cost of communications between agents.

Table 13 illustrates the average execution time (of 25 runs) of the tested algorithms on the 3-objectives DTLZ1 problem.

The lowest execution time was recorded for acMaMaPSO while acMaPSO is better than other algorithms except acMaMaPSO.

* Influence of the number of nodes on the execution time

To have an idea about the scalability of the algorithms and their convergence times with respect to the number of nodes, Fig. 10

studies the influence of the number of used nodes on the execution time.

Fig. 10 shows that increasing the number of agents leads to a high increase in the execution time. This augmentation is more excessive for the acMaMaPSO compared to other algorithms.

* Influence of the size of the initial population on the execution time

Each individual in the acMaPSO population is identified by a particle (or an agent in acMaMaPSO) representing a deployment solution that reflects a specific distribution of all nodes. Fig. 11 illustrates the variation of the runtime of the algorithms on DTLZ1 with 30 independent runtime with different population sizes.

Fig. 11 shows that the increase of the execution time is proportional to the population size. Moreover, the acMaMaPSO execution time increases faster than other algorithms, and acMaPSO is better than NSGA-III, MOEA/DD and Two_Arch2 for a population of 700 Individuals.

* Testing the effect of the number and the interdependence between objectives

Fig. 12 investigates the influence of varying the number and the interdependence of objectives on the execution time of the algorithms.

From Fig. 12, the number of objectives and the dependencies between them greatly influence the execution time: we notice that acMaMaPSO is always the least efficient. On the other hand, the behavior of the algorithms is similar for three independent and five dependent objectives (with higher values).

* Discussion

- The execution time of the acMaPSO is better than NSGA-III and Two_Arch2 but MOEA/DD has a better execution time than acMaPSO.

- The execution time of acMaMaPSO is high compared to other algorithms: in acMaMaPSO, we lose more execution time than acMaPSO but we gain the performance and the quality of the solution (although acMaPSO behavior is not so far from acMaMaPSO: see Figs. 6-9). This runtime cost of acMaMaPSO is due to the time allocated to the communication between agents. Generally, MAS simulation is, by nature, costly in terms of execution time. This can lead to a lengthy optimization process. Hence, scalable optimization of MAS must be carefully taken to ensure the speed of simulations. We use the Jade platform, changing this platform (by jaded for example) is to be considered as a perspective in order to enhance the execution time.

- Moreover, a comparison based on a theoretical runtime analysis (such as in [44]) is very interesting to analyze the time complexity of evolutionary algorithms.

6. Conclusion and perspectives

In this paper, we proposed two new variants of the PSO algorithm: the acMaPSO which includes a new concept of bird's accents and the acMaMaPSO which represents the hybridization of the acMaPSO with MAS to make the particles more intelligent, autonomous and able to learn. The proposed bird accent concept is based on the singing experience of each particle which contributes to enhancing the diversity of the population and avoiding local optima. An extensive comparison of the performance of the proposed algorithms with other ones is achieved based on HV and IGD metrics, then network metrics. Afterward, a set of statistical tests followed by an analysis of the execution time are performed. The two proposed algorithms achieve (and surpasses for specific evaluation metrics such as the number of neighbors), the performance of the standard PSO and the NSGA-III algorithm. As an application, we proposed to assess the performance of the suggested algorithms on

Table 12

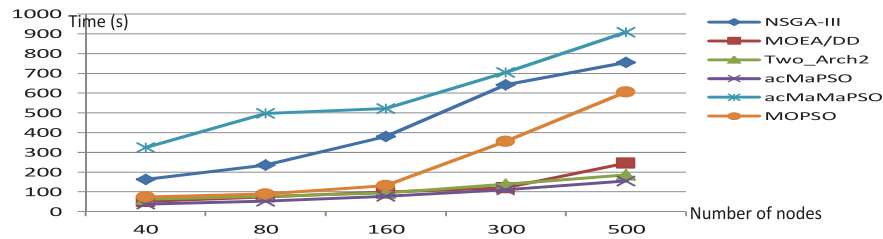
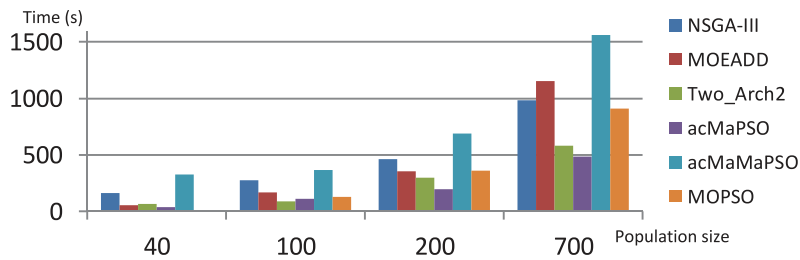
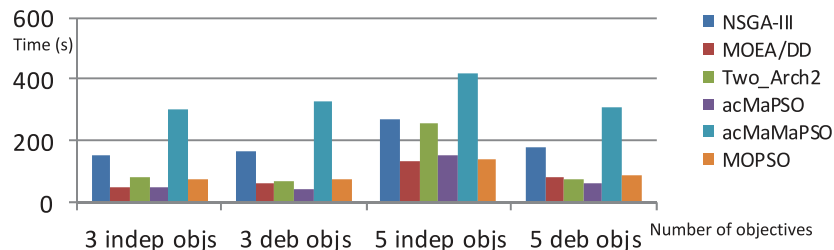
Comparison of the computational complexity on an M-objective DTLZ1 problem.

NSGA-III	MOEA/DD	Two_Arch2	acMaPSO	acMaMaPSO	MOPSO (only for $M \leq 3$)	Two-stage MaOPSO: MaOPSO/2s-pccs [28]
$O(Mn^2)$	Not mentioned by authors	$O(Mn^2)$	$O(Mn \log n)$ Worst case $O(Mn^2)$	$O(Mn^2)$	$O(M \log n)$	$O(M(n_1 + L^2 n_2)) \cong O(Mn^3)$ $n_1 n_2$ are the population sizes at stages I and II, L is the archive bounded-size

Table 13

Comparison of the average execution time on a 3-objective DTLZ1 problem.

NSGA-III	MOEA/DD	Two_Arch2	acMaPSO	acMaMaPSO	MOPSO	MaOPSO/2s-pccs{20}
1.64e+02	5.22e+01	6.37e+01	3.82e+01	3.25e+02	7.44e+01	-

**Fig. 10.** Execution time of the algorithms with respect to the number of nodes.**Fig. 11.** Execution time of the algorithms with respect to the population size.**Fig. 12.** Execution time of the algorithms with respect to the number of objectives.

the problem of 3D indoor deployment of a DL-IoT with a real-world deployment experiment based on prototyping on real nodes of an OpenWiNo-based testbed. Nevertheless, different improvements can be proposed for this study. First, as deduced from results, acMaMaPSO suffers from a high execution time. Hence the importance of proposing a less time-consuming version of this algorithm. Moreover, although our Ophelia testbed is more realistic than a platform with a large number of uniform nodes such as IoTLab [3], SmartSantander [4] or INDRIYA [5]. These latter platforms allow scaling up and testing our approach with a greater number of nodes (up to 1024 nodes). Since the IoTLab allows to test the same metrics of our experiments (RSSI, link quality...etc.), tests on this latter are envisaged in future works to prove the scalability of our approach and to compare its results with Ophelia ones.

Declarations of interest

None.

References

- [1] Zorzi M, Sepulchre R. Ar identification of Latent-variable Graphical models. *IEEE Trans Automat Control* 2016;61(9):2327–40. <http://dx.doi.org/10.1109/TAC.2015.2491678>.
- [2] Zorzi M, Chiuso A. Sparse plus Low rank network identification: A nonparametric approach. *Automatica* 2017;76(2):355–66. <http://dx.doi.org/10.1016/j.automatica.2016.08.014>.
- [3] Fleury E, Mitton N, Noel T, Adjith C. FIT IoT-LAB: The Largest IoT Open Experimental Testbed, *ERCIM News*. ERCIM; 2015, p. 4.
- [4] Sanchez L, Muñoz L, Antonio J, Pablo Sotres G, Santana RJ, Gutierrez V, et al. SmartSantander: IoT experimentation over a smart city testbed. *Comput Netw* 2014;61:217–38. <http://dx.doi.org/10.1016/j.bjp.2013.12.020>.
- [5] Doddavenkatappa M, Chan MC, Ananda AL. Indriya: A low-cost, 3D wireless sensor network testbed. In: Korakis T, Li H, Tran-Gia P, Park HS, editors. *Testbeds and research infrastructure. Development of networks and communities, tridentCom. Lecture notes of the institute for computer sciences, social informatics and telecommunications engineering*, vol. 90, Berlin, Heidelberg: Springer; 2011, http://dx.doi.org/10.1007/978-3-642-29273-6_23.
- [6] Vlado H, Andreas K, Andreas W, Adam W. TWIST: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks. In: 2nd

- international workshop on Multi-hop ad hoc networks: from theory to reality (REALMAN '06). Florence, Italy: ACM; 2006, p. 3–7. <http://dx.doi.org/10.1145/1132983.1132995>.
- [7] Mansoor U, Ammari HM. Coverage and connectivity in 3D wireless sensor networks. In: *The art of wireless sensor networks, Vol 2: Advanced topics and applications*. first ed.. Springer-Verlag Berlin Heidelberg; 2014, p. 273–324. <http://dx.doi.org/10.1007/978-3-642-40066-7>.
 - [8] Banimelhem O, Mowafi M, Aljoby W. Genetic algorithm based node deployment in hybrid wireless sensor networks. *Comm Netw* 2013;5:273–9. <http://dx.doi.org/10.4236/cn.2013.54034>.
 - [9] Corbera S, Olazagoitia JL, Lozano JA. Multi-objective global optimization of a butterfly valve using genetic algorithms. *ISA Trans* 2016;63:401–12. <http://dx.doi.org/10.1016/j.isatra.2016.03.008>.
 - [10] Daping H, Portilla J, Riesgo T. A 3D multi-objective optimization planning algorithm for wireless sensor networks. In: 39th annual conference of the IEEE industrial electronics society. 2013, p. 5428–33. <http://dx.doi.org/10.1109/IECON.2013.6700019>.
 - [11] Ko AHR, Gagnon F. Process of 3D wireless decentralized sensor deployment using parsing crossover scheme. *Appl Comput Inform* 2015;11(2):89–101. <http://dx.doi.org/10.1016/j.aci.2014.11.001>.
 - [12] Alia OM, Al-Ajouri A. Maximizing wireless sensor network coverage with minimum cost using harmony search algorithm. *IEEE Sens J* 2017;17(3):882–96. <http://dx.doi.org/10.1109/JSEN.2016.2633409>.
 - [13] 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. In: *IEEE international conference on internet of things, embedded systems and communications*. Gafsa, Tunisia; 2017, p. 20–2. <http://dx.doi.org/10.1109/IINTEC.2017.8325906>.
 - [14] Kolomvatsos K, Hadjijeftymiades S. On the use of particle swarm optimization and kernel density estimator in concurrent negotiations. *Inform Sci* 2014;262:99–116. <http://dx.doi.org/10.1016/j.ins.2013.10.025>.
 - [15] Wu C, Li H, Wu L, Wu Z. A multi-agent particle swarm optimization for power system economic load dispatch. *J Comput Comm* 2015;3:83–9. <http://dx.doi.org/10.4236/jcc.2015.39009>.
 - [16] Ahmad R, Lee YC, Rahimi S, Gupta B. A multi-agent based approach for particle swarm optimization. In: *International conference on integration of knowledge intensive multi-agent systems*. Waltham, MA; 2007, p. 267–71. <http://dx.doi.org/10.1109/KIMAS.2007.369820>.
 - [17] Peng C, Anbo M, Chunhua Z. Particle swarm optimization in multi-agent system for the intelligent generation of test papers. In: *IEEE congress on evolutionary computation*. Hong Kong; 2008, p. 2158–62. <http://dx.doi.org/10.1109/CEC.2008.4631085>.
 - [18] Kumar R, Sharma D, Sadu A. A hybrid multi-agent based particle swarm optimization algorithm for economic power dispatch. *Int J Electr Power Energy Syst* 2011;33(1):115–23. <http://dx.doi.org/10.1016/j.ijepes.2010.06.021>.
 - [19] Ait Saadi N. Thesis: Multi-Objective Wireless Sensor Network Deployment. Pierre & Marie Curie University, EDITE of Paris, defense: March, 11th 2010.
 - [20] Yu X, Liu N, Huang W, Qian X, Zhang T. A node deployment algorithm based on van der Waals force in wireless sensor networks. *Int J Distrib Sens Netw* 2013;2013. <http://dx.doi.org/10.1155/2013>.
 - [21] Boufares N, Khoufi I, Minet P, Saidane L, Ben Saied Y. Three dimensional mobile wireless sensor networks redeployment based on virtual forces. In: *International wireless communications and mobile computing conference*. Dubrovnik; 2015, p. 563–8. <http://dx.doi.org/10.1109/IWCMC.2015.7289145>.
 - [22] Mohamed Y, Kemal A. Strategies and techniques for node placement in wireless sensor networks: A survey. *Ad Hoc Netw* 2008;6(4):621–55. <http://dx.doi.org/10.1016/j.adhoc.2007.05.003>.
 - [23] Nazrul Alam SM, Haas ZJ. Topology control and network lifetime in three-dimensional wireless sensor networks. *CoRR* 2006;abs/cs/0609047.
 - [24] Kennedy J, Eberhart R. Particle swarm optimization. In: *IEEE international conference on neural networks*, Vol. 4. Perth, WA; 1995, p. 1942–8. <http://dx.doi.org/10.1109/ICNN.1995.488968>.
 - [25] Figueiredo EMN, Ludermitr TB, Bastos-Filho CJA. Many objective particle swarm optimization. *Inform Sci* 2016;374:115–34. <http://dx.doi.org/10.1016/j.ins.2016.09.026>.
 - [26] Díaz-Manríquez A, Toscano G, Barron-Zambrano JH, Tello-Leal E. R2-based multi/many-objective particle swarm optimization. *Comput Intell Neurosci* 2016;1898527. <http://dx.doi.org/10.1155/2016/1898527>.
 - [27] Britto A, Pozo A. I-MOPSO: A suitable PSO algorithm for many-objective optimization. *Brazilian symposium on neural networks*. Curitiba; 2012, p. 166–71. <http://dx.doi.org/10.1109/SBRN.2012.20>.
 - [28] Hu W, Yen GG, Luo G. Many-objective particle swarm optimization using two-stage strategy and parallel cell coordinate system. *IEEE Trans Cybern* 2017;47(6):1446–59. <http://dx.doi.org/10.1109/TCYB.2016.2548239>.
 - [29] Oesel A, Fries AC, Miller L, Gibbs HL, Soha JA, Nelson DA. High levels of gene flow among song dialect populations of the puget sound white-crowned sparrow. *Ethology* 2017;123:581–92. <http://dx.doi.org/10.1111/eth.12632>.
 - [30] Rajesh K, Devendra S, Abhinav S. A hybrid multi-agent based particle swarm optimization algorithm for economic power dispatch. *Int J Electr Power Energy Syst* 2011;33(1):115–23. <http://dx.doi.org/10.1016/j.ijepes.2010.06.021>.
 - [31] While L, Hingston P, Barone L, Huband S. A faster algorithm for calculating hypervolume. *IEEE Trans Evol Comput* 2006;10(1):29–38. <http://dx.doi.org/10.1109/TEVC.2005.851275>.
 - [32] Riquelme N, Von Lücken C, Baran B. Performance metrics in multi-objective optimization. In: *2015 latin American computing conference (CLEI)*. Arequipa; 2015, p. 1–11. <http://dx.doi.org/10.1109/CLEI.2015.7360024>.
 - [33] Shi Y, Eberhart R. Empirical study of particle swarm optimization. In: *Congress on Evolutionary Computation*, Vol. 3. Washington, DC; 1999, p. 1950. <http://dx.doi.org/10.1109/CEC.1999.785511>.
 - [34] Jason T, Shanchieh Y, Raghuvver R, Ferat S. Optimal topologies for wireless sensor networks. *Int Soc Opt Eng* 2004;192–203. <http://dx.doi.org/10.1117/12.578518>.
 - [35] Deb K, Thiele L, Laumanns M, Zitzler E. Scalable test problems for evolutionary multiobjective optimization. In: Abraham A, Jain L, Goldberg R, editors. *Evolutionary multiobjective optimization*. Advanced information and knowledge processing, London: Springer; p. 105–45. http://dx.doi.org/10.1007/1-84628-137-7_6.
 - [36] Coello CA, Van Veldhuizen DA, Lamont GB. *Evolutionary algorithms for solving multi-objective problems*. second ed.. Springer US; 2007. <http://dx.doi.org/10.1007/978-0-387-36797-2>.
 - [37] The Arduino platform (2017); Software available at URL: <https://www.arduino.cc/en/main/software> (Accessed: January 14th, 2018).
 - [38] Van Den Bossche A, Dalce R, Val T. OpenWiNo: An open hardware and software framework for fast-prototyping in the IoT. In: *23rd international conference on telecommunications (ICT)*. Thessaloniki, Greece; 2016, p. 1–6. <http://dx.doi.org/10.1109/ICT.2016.7500490>.
 - [39] Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans Evol Comput* 2013;18(4):577–601. <http://dx.doi.org/10.1109/TEVC.2013.2281535>.
 - [40] Li K, Deb K, Zhang Q, Kwong S. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Trans Evol Comput* 2015;19(5):694–716. <http://dx.doi.org/10.1109/TEVC.2014.2373386>.
 - [41] Wang H, Jiao L, Yao X. Two_Arch2: An improved two-archive algorithm for many-objective optimization. *IEEE Trans Evol Comput* 2015;19(4):524–41. <http://dx.doi.org/10.1109/TEVC.2014.2350987>.
 - [42] Figueiredo EMN, Araújo DRB, Filho CJAB, Ludermitr TB. Physical topology design of optical networks aided by many-objective optimization algorithms. In: *5th brazilian conference on intelligent systems (BRACIS)*. Recife; 2016, p. 409–14. <http://dx.doi.org/10.1109/BRACIS.2016.080>.
 - [43] Friedman M. A comparison of alternative tests of significance for the problem of m rankings. *Ann Math Stat* 1940;11:86–92. <http://dx.doi.org/10.1214/aoms/1177731944>.
 - [44] Zhou Y, He J. A runtime analysis of evolutionary algorithms for constrained optimization problems. *IEEE Trans Evol Comput* 2007;11(5):608–19. <http://dx.doi.org/10.1109/TEVC.2006.888929>.